BIT-LET

ötlet SZÁMÍTÁSTECHNIKAI MELLÉKLET · 1986 JÚLIUS 34



Múltkoriban hosszasan vitatkoztunk erről egyik barátommal. Azaz, hogy akik diákkorukban nekiesnek a számítógépnek, s az iskolaévek alatt elég komoly programozói tudást érnek el, vagy akik már felnőttként hozzájutnak egy mikroszámítógéphez, s szabadidejükben megtanulják amit meg lehet, hová jutnak, meddig, s mi lesz belőlük? Nem tudom van-e egyáltalán ezügyben kívánatos cél. S ha igen, mi az?

Milyen lehetőségek vannak? Az egyik vélemény szerint az lenne a jó, ha aki mondjuk egy Commodore-on már „mindent" tud, annak valahogyan lehetőséget kellene teremteni, hogy továbblépjen. Mondjuk legyen lehetősége, egy PC méretű géppel megismerkedni, IBM kompatibilis gépeken programozni, megismerkedni az operációs rendszer mibenlétével, meg egyáltalán, hogy mitől professzionális egy professzionális kategóriájú személyi számítógép. A másik vélemény szerint erre semmi szükség, hiszen a személyi számítógépek legalsó kategóriájával megismerkedni, rajtuk programokat írni és futtatni, ez önmagában elegendő, s a továbblépés kevesek kiváltsága lehet csupán. Igaz, hogy a legkiválóbbak kiválasztódása jó alapelv, s akár a sportban, úgy számítástechnikában is csak nyerhet a szakma, ha sok alapképzettséggel rendelkező fiatal közül választódik ki, hogy kiből lesz profi, s ki marad meg amatőrnek. Ez igaz, csak az a kérdés, hogy valóban megtörténik-e ez a kiválasztódás, valóban a legjobbak jutnak-e a nagyobb gépek közelébe, valóban belőlük lesznek-e profik vagy inkább az számít, hogy kinek milyen a hátországa, kinek milyenek a családi és egyéb kapcsolatai? Nyilvánvaló, hogy egyértelmű válasz nincs a dologra, igaz az is, hogy a legjobbak bejutnak az egyetemekre, s ha akarják profi lesz belőlük, s igaz az is, hogy az ismerkedésben, a továbblépésben sokat számítanak a személyes adottságok, kapcsolatok is.

Azon azonban érdemes lenne elgondolkodni az illetékeseknek, hogy hogyan lehetne intézményes lehetőségeket teremteni legalább a középiskolák legjobban programozó diákjai számára, hogy ha már igazán jók a mikrogépeken, legalább az alapismerkedés erejéig továbbléphessenek. Hogy nem lehetetlen amit fölvetek, azt bizonyítják a már meglévő példák. Budapesten már most is létezik olyan gépet gyártó cég, amely rendszeresen lehetőséget ad budapesti középiskolák szakköröseinek, hogy gépeikkel találkozzanak, ismerkedjenek, sőt azokon programokat is írhassanak. Nem azt akarom tehát mondani, hogy az oktatásügy illetékeseinek azon kellene törniük a fejüket, hogy hogyan juttassák az iskolákat professzionális PC-khez. Nem, ezt úgyis tudjuk, hogy irreális lenne. Azt azonban lassan napirendre kellene tűzni, hogy a haladókkal mi legyen, milyen módon tudják az iskolai szakkörök az ő továbblépési igényeiket is kielégíteni. valószínű, hogy a megoldás a fenti példában említetthez hasonló együttműködési rendszer lehet. Létre lehetne például hozni nagyobb területi egységek olyan szakköreit, amelyekbe a terület iskoláiból a legkiválóbbak járhatnak, s amely vendégként ugyan, de rendszeresen hozzájut a nagyobb gépekhez.

No és mi a helyzet azokkal, akik már nem iskolások? Hogyan lehetne az ő tudásuk továbbfejlesztési lehetőségeit is megteremteni? A kérdést föltettük a barátommal egymásnak, de válaszolni nem tudtunk rá. Mint hogy azt sem tudjuk, pedig érdekelne bennünket, hogy egyáltalán mi az igazság? Mi a helyzet ma? Valóban, aki úgy érzi már nagyon tudja a Commodore vagy a HT programozását, az mit tesz? Akar, tud találni lehetőséget a komolyabb gépekkel való ismerkedésre? Aki komoly gépi kódú programokat ír a C 64-esen, ült már valaha PC 10-es, 20-as előtt? Jó lenne tudni, hogy kinek mik a tapasztalatai? Jó lenne ha nemcsak a szerkesztő mondaná el kérdéseit, gondolatait minderről, hanem a tisztelt olvasók, azaz maguk az érintettek is. Leveleiket várja a szerkesztő:

Angyalosi László

BELÜLRŐL

HÍROLDAL

Villanyhálózaton

Angol mérnökök olyan készüléket fejlesztettek ki, melynek segítségével az elektromos hálózat vezetékrendszere is alkalmassá válik egy épületen belüli adatátvitelre, azaz a számítógépek, terminálok és perifériák közötti adatforgalom biztosítására. Az alkalmazott jelátvitel kisfeszültségű és nagyfrekvenciájú jelei a hálózati áramra szuperponálódva haladnak tovább. Az adatátvitel négy egymástól független csatornán folyhat anélkül, hogy interferencia jönne létre és ugyanabban az épületben zavarnák egymást az egyes átviteli csatornák.

Csehov

Nagy munkába fogtak a Nyíregyházi Tanárképző Főiskola kutatói. Az orosz nyelv és irodalom tanszék a Magyar Tudományos Akadémia támogatásával Csehov drámáinak nyelvészeti, stilisztikai és irodalmi vizsgálatára vállalkozott. Csehov négy nagy drámájának mintegy hatvanezer szavát – természetesen az eredeti orosz nyelvűt – kódolják és a számítógépbe viszik. A számítógép ezután – a kutatók szempontjai szerint – elvégzi a szükséges rendszerezéseket, vizsgálatokat.

Rövid hullámon

Mikroszámítógépes hálózat hordozható, kábeles összeköttetés nélküli számítógépekkel. Ezt ajánlja felhasználóinak az Electronic System Technology cég. Jelenlegi kiépítettségében az angol cég rendszerében kettőszázötvenöt mikroszámítógéppel lehet a kapcsolatot biztosítani, mégpedig rövidhullámú rádió-adóvevő rendszer segítségével. Az új rendszermegoldás egyetlen hibája, hogy az ára kétszerese a hagyományos vezetékes összeköttetésű számítógéprendszernek.

Bunker

Bomba és földrengésbiztos bunkerrendszert építenek ki a New York-i Világkereskedelmi Központ alatti sziklákban. Ez lesz a világ egyik legbiztosabb számítógépi programőrző bunkerrendszere. Feltételezve, hogy nem lesz olcsó dolog ebben a raktárban őriztetni a cégek létfontosságú programjainak másolatait, mégis sokan meg fogják azt fizetni. Ma már a hatalmas vállalatóriások léte függ számítógépes programjaiktól. Egy bármilyen katasztrófa során megsemmisülő rendszer az egész vállalat csődjét jelentené. Ezért van szükség a drága, de biztonságos raktárak kiépítésére.

KIT!

A dallasi ADTEK cég az IBM AT számítógépével kompatibilis, SERIES 286 AT típusú házilag összeszerelhető, kit formában forgalmazott számítógépét ajánlja vásárlóinak. A vásárló külön-külön megveheti a sasszét, a főkártyát, a kábeleket, a lemezvezérlőt, a billentyűzetet és más áramköri kártyákat. Miután egy szombat délután összeszerelte ezeket, hozzáillesztheti saját monitorát és lemezegységeit. Persze, hogy mit vásárol meg az egyes egységekből, az a felhasználó igényétől függ. Mindenesetre így olcsóbban jut a géphez és még hobbyjának is áldozhat.

Borsó

Nem volt falra hányt borsó a számítástechnikai propaganda a Nyíregyházi Konzervgyár részére termelő mezőgazdasági nagyüzemekben. Ugyanis számítógép irányításával végezték a borsó vetését. A számítógép a betáplált adatok – mint például a vetőmag fajtája, a termelők talaj és éghajlati viszonyai, gépi kapacitása, a konzervgyár fogadókészsége stb. – alapján állapítja meg, hogy melyik termelőegységnek mikor kell elkezdenie a vetést, hogy a gyár folyamatosan kapja majd a borsószállítmányokat.

Asszociatív

Az emberi agy előnye a nálánál sokkal nagyobb műveleti sebességű számítógépekkel szemben, hogy asszociatív tulajdonságú. Most azonban úgy látszik a számítógépmemóriák asszociativitása irányába is halad a fejlődés. A nyugatnémet Güntner Palm egy asszociatív memóriachip kidolgozásáért kapta meg a SEL alapítvány huszonötezer márkás díját. Az új rendszerű chip bevezetése valószínűleg új fejezetet nyit majd a számítógép generációk fejlődésében is.

Sárgabarack!

A Számítástechnika Alkalmazási Vállalat a SZÁMALK már eddig is több társulást hozott létre, hogy minél eredményesebben szolgálhassa a hazai felhasználókat. A legújabbat a Skála-Cooppal alakították ki. Viszonylag olcsó, nagyteljesítményű mikroszámítógépekhez kívánják juttatni a magyar vállalatokat.

A társaság a VT Computers magyar–angol vegyes vállalattól vásárolt Apricot (sárgabarack) típusú mikroszámítógépek forgalmazását kezdi meg Magyarországon, amelyekhez az alkalmazói szoftverekről, a szervizről a SZÁMALK gondoskodik.

Renoir

Ezúttal nem a nagy francia festő képei, hanem a hangja jelenti a szenzációt. Japán akusztikusok ugyanis számítógép segítségével kialakították és magnószalagon rögzítették az 1919-ben meghalt művész hangját. A művelet elektronikus hangszimulációra és anatómiai mérésekre épült. Ehhez felhasználták a művész orr-garat részéről meglévő ismereteiket, mivel ez nagymértékben befolyásolja a képzett hang jellemzőit.

Megunták?

Amerikai kutatók felmérést végeztek általános iskolás gyerekek között és a korábbiaktól eltérő tapasztalatokat szereztek. Az iskolai tananyagba szervesen beépült számítástechnikai oktatásban résztvevő tizenhárom éves gyerekek között egyre többen vannak olyanok, akik nem szeretnek a számítógéppel foglalkozni. A megkérdezettek nagy százaléka nem kíván a jövőben, majdani munkahelyén számítógéppel dolgozni, és több mint felük nem is sejtette, hogyan működik a számítógép.

Extra szuper

A számítógépek ötödik generációjához tartozó, extra szuper, gyors számítógép kifejlesztésén dolgoznak a Szovjet Tudományos Akadémia leningrádi tudományos-műszaki egyesülésében. A tervek szerint az extra

szupergép sebessége másodpercenként 10 milliárd művelet lesz. Emellett alkalmas lesz párhuzamos működésre, képi és beszédinformáció érzékelésére. Makszim Aleszandrov professzor, – az egyesülés vezérigazgatója – szerint az új gép létrehozása a 90-es évek elejére várható.

Képernyő!

Hárommillió képpontot tartalmaz az a kisméretű, színes folyadékkristályos képernyő, amit a japán Toshiba cég munkatársai fejlesztettek ki. Fényereje csaknem eléri a hagyományos megoldású képernyőkét. Az új eszközt máris alkalmazzák a számítástechnikai rendszerekben.

Märklin

Új korszak köszönt a vasútmodellezésre. A Märklin és Cie Kft. nyugatnémet vállalat szakemberei, olyan elektronikus egységet hoztak piacra, amely lehetővé teszi, hogy a vasútmodellekkel és hobby számítógéppel egyaránt rendelkezők összekapcsolhassák azokat. Az így létrejövő játékrendszerből sokkal többet lehet kihozni, mint a hagyományos vasútmodellekből. Az NSZK háztartásaiban mintegy hárommillió modellvasút és csaknem kétmillió számítógép működik. A Märklin cég egyelőre évi öt-hatezer összekapcsoló elektronikus egység eladását tervezi.

Modern

Új hajlékonylemezes tárolót fejlesztettek ki és gyártanak a MOM dunaújvárosi gyárában. Az új eszköz kapacitása kétszerese, mérete pedig fele a korábbinak és ugyanakkor cserszabatos vele. A gyár évente 40–50 százalékkal növeli termelését. Az idén huszonnyolcezer tárolót gyártanak. Az elkészült termékek jelentős része külföldre kerül. Legnagyobb külföldi vásárlójuk az NDK-beli Robotron cég.

Még Új – Országos C Egyesület

Gyerekek, felnőttek! Örsök, rajok, KISZ-alapszervezetek! Iskolák, tanszékek, klubok! Művelődési házak, kultúrotthonok! Gazdasági munkaközösségek, egyéb vállalkozások! Vállalatok!

ITT A COMMODORE EGYESÜLET!

Mit nyújt Önnek az Egyesület?

Az attól függ, hogy Ön melyik egyesületi páholyba kíván belépni. Amit mindenképpen megkap az egyesülettől, az a naprakész informáltság Commodore ügyekben, némi bepillantás a jövőbe, s lehetőség a következő páholyba való átlépésre.

COMMODORE DEÁKPÁHOLY

- negyedévenként információs bulletin friss Commodore hírekkel, új, itthon is kapható programok és könyvek listájával
- pötyögő szolgálat, amely megkíméli a tagokat a pötyögéstől
- ingyenes apróhirdetési lehetőség a Commodore újságban
- különböző vásárlási kedvezmények.

TAGSÁGI DÍJ: évente 116 forint, fél évre 60 forint

COMMODORE PLUSZPÁHOLY

- minden az előbbiekben felsorolt szolgáltatás
- a havonta megjelenő 36 oldalas Commodore Újság a Commodore Computing International és a nyugatnémet Data Welt lapokkal együttműködésben készül
- havi 50 forintos vásárlási kedvezmény a Novotrade 2C boltjaiban
- szervizkedvezmények, és egyéb pénzkímélő akciók

TAGSÁGI DÍJ: évente 1264 forint, félévre 650 forint

COMMODORE SZUPERPÁHOLY

- minden az előzőekben felsorolt szolgáltatás
- havonta 10+4 példány a Commodore újságból
- meghatározott, havonta változó cikkekből 10% árengedmény a Novotrade 2C boltjaiban

TAGSÁGI DÍJ: évente 12 128 forint, félévre 6100 forint

Az Országos Commodore Egyesület igyekszik majd tagjai részére tanácsadó szolgálatot is létrehozni és ellátni a Commodore felhasználók érdekvédelmi tevékenységét is.

HOGYAN LEHETNE ÖN IS EGYESÜLETÜNK TAGJA?

Befizeti az Ön által kiválasztott páholy tagdíját a **Budapest XIII., Visegrádi u. 7/b** alatti OTP fiókban vezetett MNB 217–98292, OTP 565–3610 számú számlára. Ezt megteheti személyesen vagy postán. Rövid időn belül megkapja tagsági igazolványát, és az első Commodore újságot.

Pontozott pontozók

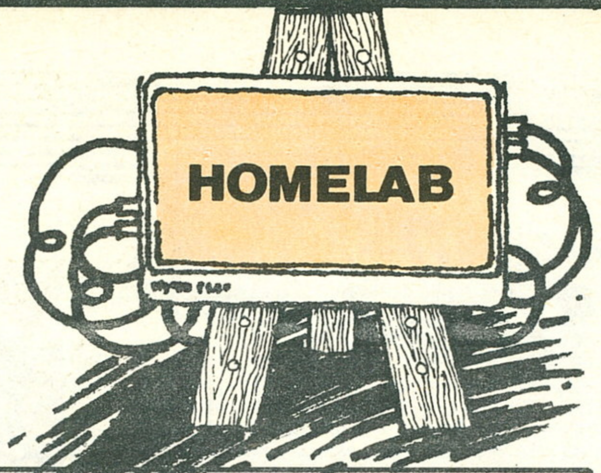
Magyar szakemberek a tornászbajnokságok pontozóbíróinak pontozására, rangsorolására készítettek számítógépes programot. A számítógép a versenyek valamennyi pontszámát rögzíti, jelzi, hogy a pontozók mennyire térnek el az érvényes pontszámoktól, megállapítja, hogy a pontozóbírók elfogultak-e valamelyik ország vagy egyes tornászok iránt. A számítógép által megállapítottakat, és a pontozók rangsorolását a nemzetközi szövetség vezetői bizalmasan kezelik.

Kormos gép

Ha a gép nem is, de a téma igen. A Baranya megyei Kéményseprő és Tüzeléstechnikai Szolgáltató Vállalat egy Commodore 64 típusú mikroszámítógéppel megkezdte a pécsi kéménykataszter elkészítését. Számítógép segítségével állítják majd ki a kéményseprési számlákat, és már azzal számítják ki a gázkémények terhelhetőségét.

Hangfelismerő

Az amerikai The Voice Connection cég Intro Voice IV. néven egy hangfelismerő egységet fejlesztett ki az IBM PC XT részére. Az egység segítségével a számítógép mintegy ötszáz, a felhasználó által meghatározott, kimondott szóra, illetve kifejezésre válaszol. A rendszer felismeri a tipikus szavak több mint kilencvennyolc százalékát, kevesebb mint 0,2 másodperc alatt.



VALLATÓ HOZZÁSZÓLÁS

Nem tsz – szövetkezet!

A BIT-LET májusi számában leközölt iskolaszámítógép vallató néhány sorához lenne pár gondolatom, amelyeket szeretném, ha legközelebbi „BIT-LET" olvasói posta rovatában megemlítenének.

1. A HOMELAB 4 számítógépet nem a dombóvári tsz, hanem Szövetkezetünk gyártja, ami egy egész egyszerű tesztmódszerrel a gépkönyv kinyitásával megállapítható.
2. A hazai szövetkezeti ipar is produkált már világszínvonalú termékeket, ugyanakkor a nagyipar is gyártott már bóvlit.
3. Nem túl szerencsés, ha egy pártatlan bíráló a gyártó hovatartozásából eredő előítélettelindul neki egy termék minősítésének.

Tóth Attila, a dombóvári COLOR Ipari Szövetkezet elnöke

A gyártó kilétét illető tévedésért elnézést kérünk. A „Sajnos a gép mechanikai kivitele még mindig magán viseli a »tsz-gyártmányok« jellegzetes hibáit." – megállapításunkat továbbra is fenntartjuk, s nem előítéletnek, hanem tapasztalati ténynek véljük.

A Homelabról mégegyszer

A májusi számban megjelent „Kinpadon a jövő iskolaszámítógépe" című cikkhez szeretnék hozzászólni, elsősorban a Homelab 4 számítógép védelmében. Természetesen nem azzal a céllal, hogy bebizonyítsam: ez a gép a legalkalmasabb az általános-, és középiskolák számára (bár ha esetleg ilyen következtetésre jut az olvasó, az az ő „hibája"). Csupán néhány tekintetben szeretném kiegészíteni a kényszerűen rövid valló megállapításait. Tapasztalataimat saját építésű gépem egy éves üzeme alatt szereztem, így azok mélységben nyilván nem összemérhetők az említett cikkben leírtakkal és belső érzelmektől sem mentesek, mindazonáltal nem szeretnék ezzel az előnnyel visszaélni.

Elöljáróban hadd tegyek néhány általános jellegű észrevételt. Bebizonyosodott, hogy nyolc óra kevés hat gép érdemi összehasonlítására, főleg, ha az inkvizítorok nagy részüket nem ismerik. Ennek nyilvánvaló kivihetetlensége tükröződik a Vallató sok helyén (pl. 5. kín: Tárolás). Ezért nem is lehet csodálkozni azon, ha a nem elég részletes vizsgálatok eredményeit szubjektív vélemények „egészítik ki", és itt előnybe kerülnek azok a gépet, amelyek jórészt már ismertek: C 16, HT, Pro-Primo. A kevés idő csupán a durva hibák (pl. a HomeLab 4 billentyűzete, a MICRODAT hiányos karakterkészlete) kiszűrésére lehetett elegendő, a finomabb részletek homályban maradtak. (Csak zárójelben jegyzem meg, hogy szerintem legalább ilyen durva hiba a számolási pontatlanság, amelyen az egyik nyertes, a C 16-os csúnyán leszerepelt. Ilyen gépet nem szabadna az iskolákban terjeszteni.)

Állításomat, miszerint a HomeLab 4 számítógép sok lényeges jó tulajdonsága fölött átsiklottak az Inkvizítorok, az egyes kínok sorrendjében igyekszem alátámasztani.

1. kín: Billentyűzet

Ehhez sok mindent nem tudok hozzátenni, a billentyűzetet én is inkább magam építettem. Érdekesség, bár talán a programnyelvhez tartozik, hogy a KEY parancs kiadása után az F1 vagy F2 gomb és egy karaktergomb egyidejű lenyomásával egy mozdulattal hívhatók elő a BASIC kulcsszavak, akár a ZX Spectrumnál, csak ami ott kényszer, az itt opció. (Az utasítások ekkor is beviheték betűnként.) A RESET gomb megnyomása után bármely futó program megáll, de nem vész el: listázható és újraindítható.

2. kín: Ékezetes betűk

Mi az, hogy ékezetes betűk? Elég-e, ha egy gomb lenyomásakor megjelenik pl. az á a képernyőn, vagy az is követelmény, hogy az
á=99: Print á
utasítássor eredménye 99 legyen és ne Sn error? Ez itt nem derül ki, pedig lényeges. Csak arról a gépről lehet állítani, hogy tudja az ékezetes betűket, amelyik a fenti sort helyesen dolgozza föl. (A HomeLab „persze" ilyen, a többi nem.)

3. kín: Periféria csatlakoztatási lehetőségek

„A HomeLab csatlakozási lehetőségei szegényesek: video és RF kimenet, mindkettő koaxiális csatlakozóval, 1 db magnócsatlakozás, busz és Centronics kimenet. Joystick nincs, és valójában a Centronics sem szabványos."

Magnócsatlakozóból szerintem egy is elég (hacsak nem sztereóban programozik valaki). A Centronics kimenet sok más géptől eltérően mindenféle elektronika nélkül, pusztán a két megfelelő csatlakozóval és a közöttük levő kábellel illeszthető bármely ilyen bemenetű nyomtatóra. Itt az inkvizítorok alighanem félreértettek valamit a szabvány körül. A HomeLab kimeneti jelei szabványosak, csatlakozója viszont nem az, de arra nincs is szabvány! A „szegényes" buszcsatlakozóra pedig tényleg szinte bármi köthető. Hogy ezt a mesét már ismerik? Akkor néhány lehetőség: EPROM-égető (2716-tól 27256-ig bármit éget), 14 bites analóg-digitál és 12 bites digitál-analóg átalakító, beszédszintetizátor (nem csak a kártya, hanem a hozzávaló szoftver is, amely „felolvassa" a programlistát, vagy a képernyőn megjelenő üzeneteket, „elmondja" a leütött billentyűt!), CP/M bővítőkártya (!). Ez utóbbi új távlatokat nyit a gép előtt, fontosságát, hasznosságát nehéz lenne túlbecsülni. (Később még lesz róla szó.) Az egyetlen, ami tényleg hiányozhat (és nincs sem a HT-n, sem a C 16-on, sem a Pro-Primón): egy soros RS–232 adatvonal, de periféria-bővítés formájában ez is megoldható.

4. kín: Képernyőkezelés és grafika

Van egy jó meg egy rossz hírem.

A jó: elkészült az 512∗256 pontos igen nagy felbontású grafika a hozzávaló szoftver segédlettel!!! (Ilyen nagy felbontása alighanem csak a TVC-nek van.) Nagy képsíkot tud kezelni, és rendelkezik 24*20 pontos sprite-mozgató hardverrel is.

A rossz: egyszínű. (Bár személyes véleményem szerint jobb egy egyszínű videómonitor, mint két színes TV, már ha programozni is akar az ember).

5. kín: Tárolás megbízhatósága

A HomeLab 4 gép kazettás tárolása merőben szokatlan. A következő mágikus jeleket kell beadni neki:

Tárolás:	SAVE „név"
Betöltés:	LOAD „név"
Ellenőrzés:	VERIFY „név"
Ciki.	

A tárolás elegendően gyors, ami részben a kiváló BASIC-ban írható tömör programoknak köszönhető. A visszaolvasás 20–30 kbájt rekordhosszig problémamentes (hosszabb adathalmazt eddig nem sikerült előállítanom). Igaz, a magnómat is saját kezűleg építettem MK 25-ös mechanikából, de a kialakult gyakorlattal ellentétben ettől a magnó egyszerűbb és olcsóbb lett.

6. kín: A gép programnyelve

Egy kiegészítés a dicsérethez: a HomeLab szintén rendelkezik egy különleges lehetőséggel: különböző programszinteken egymástól teljesen független, akár azonos sorszámú sorokat tartalmazó több különböző program lehet egyidejűleg a gépben, amelyek külön-külön listázhatók és futtathatók. Ez eddig még nem tűnik túl hasznosnak, de ha elárulom, hogy ezek a programok átugorhatnak más szintekre és szubrutinokat is hívhatnak azokról, bátran mondhatjuk a C 16-oshoz hasonlóan, hogy „...már-már a struktúrált nyelvek előnyeit rejtő lehetősége is van".

Ezenkívül lásd az Aircomp vallatóját a Szuper BIT-LETben. (A HomeLab 4 az Aircomp továbbfejlesztett változata.)

9. kín: Editálási funkciók

Érdekesek az osztályzatok. A HomeLab editálási rendszere lényegében azonos a C 16-oséval, „csupán" az erősen kifogásolt idézőjeles üzemmód hibája lett tökéletesen kijavítva. Ehhez képest a C 16 3,7-et, a HomeLab 2,6-ot kapott. Hm! Itt szeretnék megemlíteni egy apróságot. A HomeLab gépnél, szemben az összes többivel, a logikai sor legnagyobb hosszát csupán a képernyő korlátozza, tehát írhatunk akár 2000 karakteres programsort is. Bár ez a gép a leggyorsabb működésű, néha itt is spórolni kell az idővel. Ilyenkor, hála ezen tulajdonságának és magasszintű BASIC-jének, szerencsés esetben egy hosszú program is két-három programsorba összehúzható, ami lényeges sebességnövekedést eredményez. Szubjektív véleményem: a HomeLab editálásából tényleg csak a szintaktikai ellenőrzés hiányozhat, nekem még az sem. Ha lesz ilyen is a gépen, feltétlenül kikapcsolható legyen.

11. kín: Tanulhatóság

Alighanem én egy zseni vagyok, mert könnyen, gyorsan és nagy intellektuális élvezettel „tanultam meg" a HomeLabot... Lehet, hogy a gép magyar hibakiírása „sokszor nem a hiba lényegét fogalmazza meg", de egy ötödikes iskolás számára a Bad subscript error egyáltalán semmit nem fogalmaz meg.

12. kín: Emberközelség

Idézetek a már említett Aircomp-vallatóból:
Földi Péter: „Nálam bevágódott az Aircomp!"
Fodor József: „A gép oktatásra is kitűnő! Az együtteltöltött idő alatt nagyon megszerettem!"
Barabási Rezső: „Az ember–gép kapcsolat szempontjából igazán ideális."
Kovács Levente: „A gép kellemesen »puha«, kellőképpen »lezser«, ugyanakkor kötött, precíz!"

14. kín: Gépi kódú programozás lehetősége

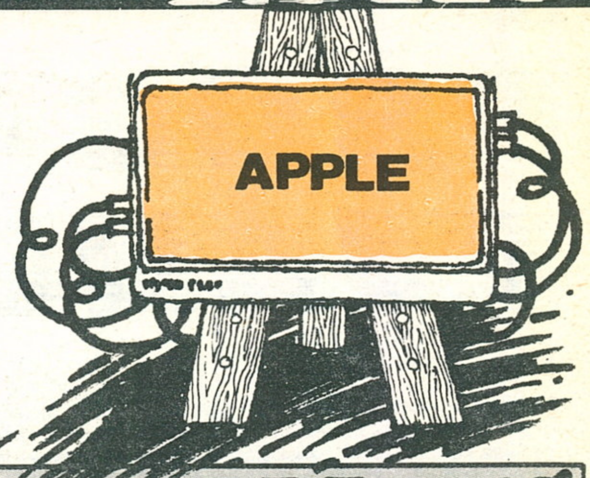
Alapkiépítésben a HomeLab és a C 16 teljesen azonos lehetőségekkel bír. De mivel az inkvizitorok a HT gépnél mindig arra hivatkoznak, hogy segédprogrammal Spectrummá alakítható, hadd tegyek én is hasonlót. Létezik a HomeLabhoz 8 kbájt EPROM bővítés formájában (amely a RAM-területből nem foglal helyet) egy nagyon okos szimbolikus ASSEMBLER/EDITOR/DEBUGGER (továbbá ebben van az EPROM-égető szoftverje). Ezt tehát nem szalagról kell beolvasni, megbízhatóan rendelkezésre áll, egyetlen utasítással aktivizálható. Ilyen szinten ez tökéletes megoldás, de ha valaki erre sem ad ötöst,... az olvassa el a levél végét!

16. kín: A memória mérete

Értem én a gondokat, de a táblázat szerint mégis-csak sikerült a szükséges titkos adatokat beszerezni. Továbbá ha már valaki osztályoz, akkor a táblázat dacára miért ad a TVC-nek és a HomeLabban 4-est, a Pro-Primónak 4/5-öt és a HT-nek 5-öst? Még annyit, hogy a 3. kínnál említett nagyfelbontású grafika alaphelyzetben nem foglal helyet a 48883 byte szabad területből, egyébként pedig a RAM-mező bármely 16 kbyte-os szeletére definiálható. Ugyanígy nem zavarja a BASIC RAMot a képernyő-RAM és a billentyűzetmező sem a szellemes memórialapozó hardvernek és szoftvernek köszönhetően. Ha mégis lett volna osztályzás, a HomeLab ötöst érdemelt volna.

18. kín: Szoftverellátottság

A HCC HomeLab szekciója tudomásom szerint eddig négy szoftver-füzetet adott ki a géphez, tele BASIC és gépi kódú programokkal, amelyek természetesen kazettán is hozzáférhetők. A már említett szimbolikus ASSEMBLER/EDITOR/DEBUGGER mellett létezik a géphez fig-FORTH 1.1-es compiler, és tudvalevőleg a FORTH programok a nyelv szerkezete miatt könnyen átvihetők egyik gépről a másikra. Aki ezt a választékot kevésnek találja (tényleg nem túl nagy) várja ki a levél végét!



VALLATÓ HOZZÁSZÓLÁS

Végül

Azt hiszem, sikerült annyit bebizonyítanom, hogy a HomeLab 4 alaposabb, részletesebb vizsgálatnak alávetve (amely még most sem késő) talán nem is tűnik olyan rossznak (lásd a már sokszor citált Aircomp-vallatót), és ez már eredmény. De még nem árultam el mindent! Lássuk a már beharangozott meglepetést!

A befejezés bevezetéséhez érdemes elolvasni a Mikroszámítógép Magazin 1985/1-es számának 12. oldalán kezdődő „Nyolcbites mikroszámítógépek operációs rendszerei" című cikket. Ebből kiderül, hogy milyen hatalmas előny, ha egy gép alkalmas a CP/M operációs rendszer futtatására, egycsapásra kiváló programok tucatjai, ha nem százai közül választhat a felhasználó. Nos, a HomeLab 4 rendelkezik CP/M rendszerbővítéssel, amellyel a gép tulajdonképpen „magasabb osztályba" lépett a viszonylag alacsony ár megtartásával. (A „tulajdonképpen" szót majd akkor lehet elhagyni, ha a gép külcsíne összhangba kerül a belbeccsel.)

Általános célú (pl. szövegszerkesztő, adatbázis-kezelő, stb.) programokon kívül a CP/M rendszerben elérhető valamennyi jelentős program nyelv fordítója, többnyire a 8-bites gépeken lehetséges legjobb minőségben. (Például: MicroSoft BASIC interpreter és compiler, MACRO–80 assembler, PASCAL, FORTRAN, COBOL, C, FORTH, stb.) Mindezek alapján úgy vélem, hogy a CP/M operációs rendszer elterjedése a magyar iskolákban (főleg a középiskolákban), még ha csak a BASIC fordító betöltésére használnák is eleinte a gyerekek, nagy lépést jelentene a számítástechnika magyarországi társadalmasításának útján. Az utóbbi néhány évben talán sikerült elérni, hogy a mikrogépeket már ne csak színes-hangos játékprogramok futtatására való eszközöknek tekintsék, de közelebb került a BASIC és gépi kódú programozás is a „jövő nemzedékéhez". Sokan azonban átestek a ló másik oldalára: azt hitték, a játékgépekkel minden programozási, vezérlési, adatnyilvántartási feladat ésszerűen megoldható. Éppen ezért a következő lépcsőfok a látókör kiszélesítését, az operációs rendszer, a struktúrált programozás, a számítógép-hálózat, stb. fogalmak megismertetését jelenthetné. Úgy tűnik, ebben a (remélhetőleg meginduló) folyamatban a HomeLab nem töltheti be társadalmi méretekben az indukáló forrás szerepét, de úgy tudom, kissé nehézkes hardver-felépítése dacára készül a CP/M a TVC gépre is. Szívből kívánom konstruktőreinek, hogy munkájuk mielőbb eredményes legyen abban az értelemben is, hogy a CP/M terjedjen el az iskolákban. Ebből jelentős hasznot húzhatna a TVC is, a HomeLab is, de ami sokkal fontosabb: országunk számítástechnikai kultúrája is.

Mészáros Gyula
1029 Bp., Zsíroshegyi út 110.



Az APPLE három pontja...

A májusi BIT-LET-ben megjelent az iskolaszámítógép-pályázatra érkezett gépek vallatásáról szóló beszámoló. A cikkben a Datacoop–Microdat gépről, amely egy APPLE II gép az a mondat olvasható, hogy „Tagadhatatlan, hogy a maga idején forradalmi gép lehetett, de ez az idő elmúlt...". Nos mit jelenthet ez a három pont?

E sor írója – úgy tűnik – nem tudott írni bővebbet e megjegyzéstől és így jobb híján három pontot tett, amely annyit is jelenthet, hogy nincs róla véleménye vagy egyáltalán nem is ismeri a gépet. Tény az, hogy itthon nagyon kevesen ismerik jól az APPLE rendszerű gépeket. Sokan formálták véleményüket a 8 bites gépekről a C 64 ismeretében. De a két gépet egyszerűen nem lehet egy kalapba tenni, hisz fejlesztésekor más-más volt az alkalmazás célja. A két gép közti különbség végeredményben az árában is mutatkozik (Az APPLE IIe alapgép a C 64 háromszorosába kerül). Jellemző a különbségre az is, hogy az APPLE II kivitele továbbra is amerikai engedélyhez van kötve, tehát továbbra is részleges embargó alá esik.

Hangsúlyozni szeretném, hogy semmilyen üzleti vagy baráti kapcsolatom nincsen sem a MICRODAT, sem az APPLE-céggel. Soraimat tehát egyszerű APPLE-rajongóként írom, aki szeretne véleményt nyilvánítani és ezzel néhány tévhit feloszlatásához hozzájárulni.

Véleményemet tehát három pontban próbálom kifejezni:

● Az APPLE II gép példás struktúrával rendelkezik, mely a nagyobb darabszámban gyártott 8-bites processzort alkalmazó gépek esetén szinte egyedülálló a világon. Ezt a struktúrát ma valamennyi komoly – professzionális gépeket gyártó – cég is alkalmazza, ha a széleskörű használat a cél.

Ez a struktúra teszi lehetővé a gép könnyű bővítését mindenféle perifériával és más processzorok alkalmazását is.

És ez a struktúra (HW és SW vonatkozásában) az, ami nem változott meg 1977 óta, és minden jel arra mutat, hogy a következő években sem fog. Ez nem jelenti azt, hogy maga a gép nem változott volna, hisz ma az APPLE IIe-ben is a legmodernebb alkatrészeket és – mint a többi nagy szériát gyártó cég gépe is – nagyintegráltságú úgynevezett Customer IC-t is tartalmaz. Az idén várható a JONATHAN nevű gép bemutatása, mely az APPLE II egyeneságú leszármazottja lesz és 8/16 bites processzort tartalmaz majd. Ezen a gépen egy az egyben alkalmazható lesz a legtöbb eddig az úgynevezett APPLEBUS-ra fejlesztett perifériaillesztő.

A floppyillesztés megoldása olyan olcsó, hogy a kismértékű árnövekedés a magnóval szemben bőven kompenzálható a floppy adta lényegesen nagyobb lehetőségekkel. A floppyforgalom sebessége még az IBM–PC-nél is nagyobb, tehát lényegesen gyorsabb, mint a C 64 esetén. Egy komplett floppy másolása a C 64-en a legközvetlenebb módon és minden ellenőrzést elhagyva kb. 9 percet vesz igénybe. Azonos információmennyiséget az IBM PC kb. 1 perc alatt másol. Ez a folyamat az APPLE esetén 32 másodperc (!).

Sokszor lehet hallani azt a véleményt, hogy a 6502-es processzor lassú, hisz csak 1 MHz-es órafrekvenciával jár. Nos ezzel az órajelfrekvenciával is vetekszik a Z80A-s processzort tartalmazó és 4 MHz-es órajellel működő rendszerekkel, mert amíg a Z80-nak egy memóriahely olvasásához 13 óraciklusra van szüksége, addig a 6502 ugyanazt a műveletet 4 órajel alatt végzi el. A maradék sebességkülönbség leküzdéséről a 6502-es rövidített címzési módok gondoskodnak. A memória kapacitását illetően most már 128 K a standard, de van több olyan bővítőkártya is, mely a kapacitást 128 K-tól akár 16 M-ig növeli, és ezt akár a legrégibb APPLE II-ben is.

●● Az APPLE II. dokumentáltsága és a róla rendelkezésre álló irodalom épp a gép (külföldi) széles körű elterjedése miatt olyan színvonalas és bő, hogy ez nagyon könnyíti a gép alkalmazását a mikrogépes világ szinte valamennyi területén. Nem utolsósorban ennek köszönhető a bő hardver- és szoftverválaszték a géphez.

●●● A gép szoftverellátottsága a legjobbak közé tartozik, sőt talán nincs is olyan 8-bites gép, melyhez nagyobb szoftverkínálat lenne. Ennek az az oka, hogy a legelső verziójú APPLE II gépre írt program valamennyi újabb verziójú APPLE gépen is fut (még a MACINTOSH gépeken is). Az APPLE II. gépeken nem csak a – ma tényleg elavult – APPLESOFT BASIC-ben lehet dolgozni, hanem rendelkezésre áll az igen korszerű UCSD–PASCAL, több FORTH-verzió, LOGO, PILOT, PROLOG és a megfelelő bővítőkártyák alkalmazásával a CP/M operációs rendszer teljes programválasztéka, MODULA2, sőt UNIX is.

A Z 80-as kártya alkalmazásával be lehet tölteni az MBASIC-et, amely szinte azonos a HT iskolaszámítógépen alkalmazott BASIC-interpreterrel, ill. a GBASIC-et, amely az MBASIC grafikus utasításokkal bővített változata.

Összefoglalva a három pontot nyugodtan megállapítható, hogy az APPLE II még ma is a 8-bites gépek világában a legkorszerűbbek közé tartozik. Hisz maga az APPLE cég kényszerült rá, hogy az egyik – már MACINTOSH-t kibocsátó – gyárát állítsa vissza APPLE IIe gyártására. És az sem véletlen, hogy az USA-ban az iskolákban alkalmazott számítógépek 60%-a APPLE II. Hasonló az arány Franciaországban és az NSZK-ban is (Bajor tartomány kivételével). És az sem lehet véletlen, hogy a Szovjetunióban és Bulgáriában is az APPLE rendszerű iskolaszámítógépek mellett döntöttek, sőt a Szovjetunióban tervezik egy komplett gyártósor üzembehelyezését, melyen kizárólag APPLE IIe rendszerű gépeket kívánnak gyártani.

Az APPLE II struktúrájánál és szoftverellátottságánál fogva kiválóan alkalmas lenne az informatika alapjainak elsajátítására. De úgy tűnik, hogy a hazai informatika még a közeljövőben is a magnóval használható BASIC-nél ér véget.

Ennek ellenére talán meglepőnek tűnik, ha mégsem javaslonám az APPLE rendszerű gépek elterjesztését az iskolákban. Az APPLESOFT BASIC, valamint az APPLE monitorprogramja – elavultsága ellenére – védett termék. Ez vonatkozik az összes a harmadik pontban felsorolt korszerű programrendszerre is. A gépek elterjesztése vagy a szerzői jog figyelembevételét vagy ennek megsértését vonná maga után. Ismerve a hazai szoftverhelyzetet, valószínűbb lenne az utóbbi. Ez a veszély – úgy tűnik – a többi iskolaszámítógépre pályázó computer esetén nem forog fent.

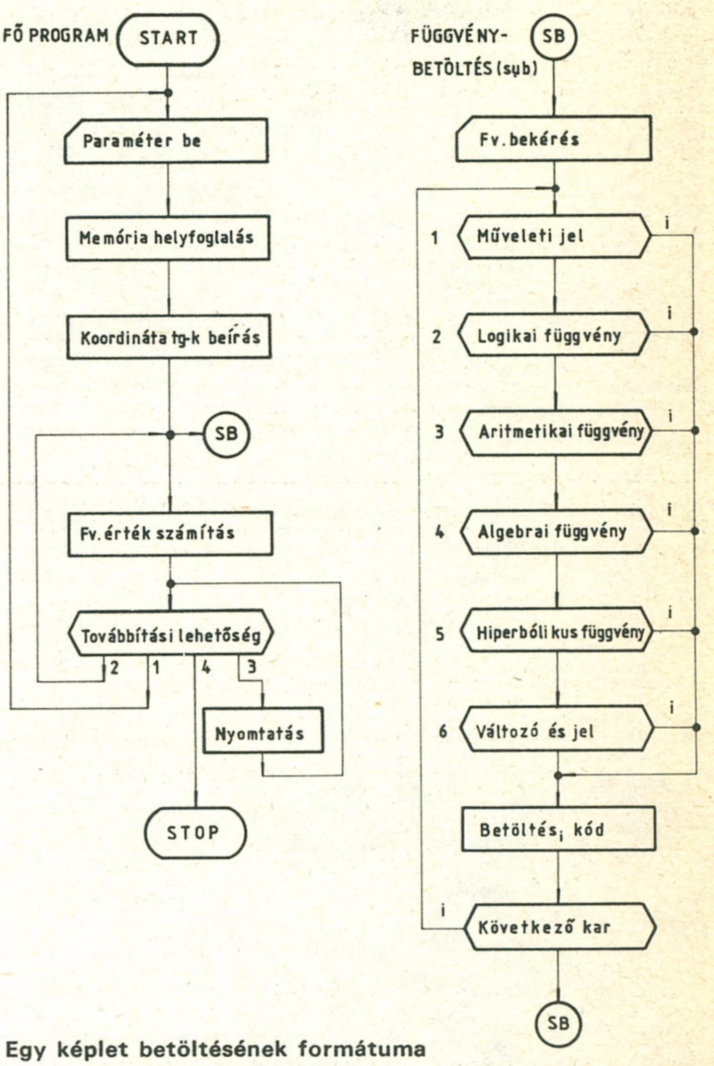
Diebel Dietrich
HCC (Számítógép Építők klubja) APPLE szekció vezetője

PROGRAM AJÁNLAT

HT 1080Z Függvény-ábrázolás

A program működése közben függvényábrázolást hajt végre a géphez kapcsolt nyomtatón. Ennek előnye, hogy aki nem tud otthon számítógépet üzemeltetni (csak az iskolában, vagy a munkahelyen fér hozzá), az is elkészítheti a függvényeit és a munkája, tanulása során ki tudja elemezni.
A printeren a nyomtatott pontok (ugyanis pontokból épül fel a függvény képe) 1/108" nagyságúak, 1/72" távolságra, gyakorlatilag egy ceruzavonal paramétereinek felelnek meg.
Egy A/4 szélességű lapon kb. 500 pont lehet. Ezen a méreten
– bármilyen függvényt
– tetszőleges nagyítás mellett lehet megrajzolni.
Természetesen nem kell az egész papír szélességét kihasználni, ha a feladat nem kívánja meg.
A nagyítást, a min. és max. x, y értékeket és a függvények egyenletét input-ként lehet megadni. A betöltés után lehetőség van újabb függvény beírására is, így görbék aszimptotáit, közelítő polinomokat, tükrözési tengelyeket együtt lehet a görbéjével megrajzolni.
A függvényábrázolás sebessége, egy A/4 méretre vonatkoztatva kb. 2 perc (HT-től ez nagyon is jó).
Ha a függvényábrázolás nem jutott el a tökéletességig hazánkban, úgy ezzel a programmal sem fog, de ha valaki nem akarja maga megírni a programot és szüksége lenne rá, azt használni tudja.
A program futtatásakor első lépésként bekéri az ábrázoláshoz szükséges paramétereket, majd ezeket átszámítja a használható értékekre. Itt kap értéket a π és az e is. (π=3,14, amit P!; e=2,71, ezt E! változónév alatt találjuk).
Második lépésben a memóriában való helyfoglalás történik meg. Első, bal felső képpont a 20010-dik byte. Innen indul a memória nullával való feltöltése, beadott paraméterekkel kiszámított értékig. Ez biztosítja hogy ne történjen fölösleges cikluslépés.
Harmadik lépésként a koordináta tengelyek beírása történik meg. Ezeken szándékosan nem akartam egységeket bejelölni, mivel a (példákból látható), nem csak függvény rajzolásra használható és zavaró lenne.
Negyedik lépés a függvény inputként történt bekérése és betöltése a meghatározott sorba, amiről később lesz szó. Ötödik lépésben a függvényértékek számítása és betöltése történik a memóriába.
Ha a számított érték nincs az értékkészletben, vagy a fv. nincs értelmezve (nullával való osztás, nem megengedett argumentum), akkor ezt az ONERROR utasítás segítségével át tudjuk lépni; nem fog megállni a program futása.
Ezután egy feltétel kiértékelésére kerül sor, hogy éppen mit szeretnénk csinálni. Itt a lehetőség:
1. Új tartományt szeretnénk értelmezni, ezzel kitöröljük az előző fv-t is.
2. Fv. beírás; itt van lehetőség újabb egyenletrendszer beírására, így több görbe egymásra rajzolására is.
3. A számított görbe (görbék) kirajzolásának módja; az utoljára beírt görbe egyenletrendszere a képernyőn és a nyomtatón is megjelenik.
4. Befejeztük a munkát.
A nyomtatási blokk három lépcsőből épül fel. A belső ciklus egy 50 bitmintából álló egységet küld ki (egyébként a legkisebb egység „50x8 bitpont", amit ki lehet nyomtatni), amelyet az abszcissza irányára annyiszor ismétlünk, míg egy sort ki nem nyomtattunk. A középső ciklus lefutása során soremelés történik, amelyet a sorok kivitelét megvalósító ciklus követ.
Előzőleg már megtörtént a soremelés beállítása úgy, hogy az új sorban a pontok megfeleljenek az egyenletes osztásnak. Ezt a blokk végén vissza is kell állítani, hogy a kellemetlen meglepetéseket elkerüljük.
Függvény betöltése. Ez egy magában is jól használható, DEFFN utasítást helyettesítő szubrutin. Igaz, a cél érdekében jól meg kellett növelni, de nagyon jól lehet használni.
A szubrutin lelke, az a gondolat, hogy a BASIC programban a változók karaktereinek ASC kódját tároljuk, a műveleti jeleknek és a függvényutasításoknak pedig a tokenjét. Ez úgy biztosítható, hogy az „input szövegből" kivágunk egy, kettő, három és négy karakteres csoportokat, amit összehasonlítunk a DATA-ban tárolt utasításokkal. Ha van olyan, amivel egyezik, akkor az utasítás kódját olvassuk a sorban következő helyre, ha nem egyezik, akkor az ASC kódot. Ilyenformán a szubrutin által nem értelmezett utasítást változónak fogja tekinteni

BLOKKDIAGRAM



Egy képlet betöltésének formátuma

BE	Y	=	3	*	SIN (	x	)	↑	2
1				*				↑	
2		=							
3									
4					SIN				
5									
6	Y		3		(·	x	)		2

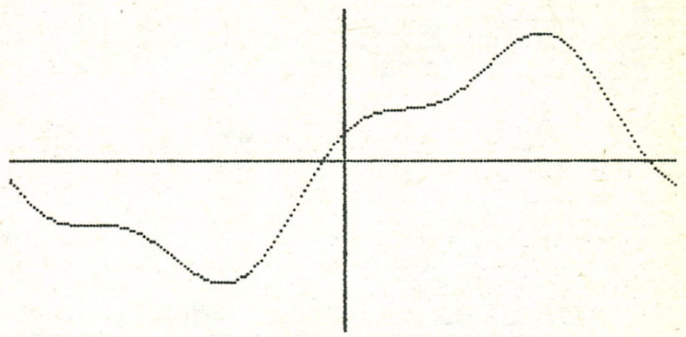
a program. Így végiglépegetve a szövegen minden karakter és utasítá a helyére fog kerülni.
A példákon látni a futtatás néhány eredményét. Természetesen nemcsak függvényt rajzolhatunk, hanem bármilyen ábrákat is, amit egyenletekkel leírhatunk és azt összemásoljuk. Ezt az OR és AND segítségével, az IF, THEN, ELSE utasításokkal végezhetjük el. Annyi módosítással, hogy az OR helyett % jelet, AND helyett pedig & jelet írunk a szövegbe. Nagyon ügyelve arra, hogy egy sorba kettő utasítás nem írható, mert az INPUT rutin csak az első jelig fogja értelmezni a szöveget.

Szabó Lajos
Budapest XI., Irinyi u. 9.

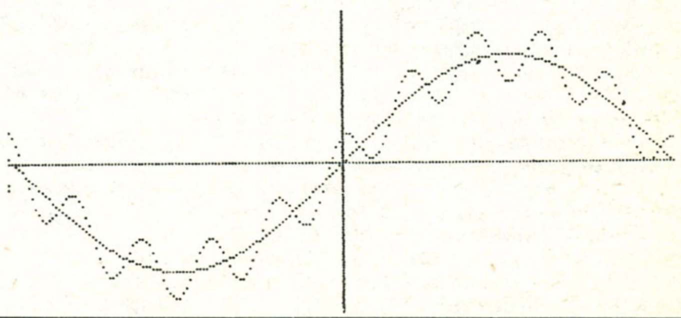
```
Szabo Lajos  1986. 02. 05.

10 CLS:CLEAR 250:DEFINT A-H
20 PRINT,"Fuggvenyrajzolo program"
30 PRINT" Uj fuggveny beiras"
40 G=0:INPUT "       min x ";G
50 H=5:INPUT "       max x ";H
60 E=1:INPUT "       max y ";E
70 F=0:INPUT "       min y ";F
80 D=1:INPUT "   nagyitas ";D
90 U=D*1.25:E=INT(U*E+1)*8:F=INT(U*F-1)*8:D=D*10:E!=EXP(1)
100 U=1/D:G=INT(G*.2+.8)*5*D:H=INT(H*.2+.8)*5*D:P!=4*ATN(1)
110 CA=20010:CC=CA+(H-G)*(E-F)*.125
120 CLS:PRINT,"Memoria helyfoglalas"
130 FOR AC=CATOCC
140 POKE AC,0
150 NEXT AC
160 A=CA+E*(H-G)*.125
170 FOR GH=GTOH
180 POKE A,128
190 A=A+1
200 NEXT GH
210 A=CA-G
220 FOR EF=ETOF STEP-8
230 POKEA,255
240 A=A+H-G
250 NEXT EF
260 GOSUB 680
270 CC=CA-G
280 ON ERROR GOTO 660
290 FOR GH=GTOH
300 T=T+U
310 X=GH*U
320         'Ide kerul a fuggveny
:
330 C=Y*D+.5
340 IF C>E OR C<F THEN 390
350 A=(E-C)*.125
360 B=2Ä(7-E+C+A*8)
370 A=A*(H-G)+X*D+CC
380 POKEA,(PEEK(A)ORB)
390 NEXT GH
400 GOTO 590
410 CLS
420 PRINT,"Nyomtatas !"
430 PRINT J$
440 LPRINT J$
450 LPRINT CHR$(27);CHR$(64)
460 A=CA
470 LPRINT CHR$(27);CHR$(65);CHR$(8);
480 FOR EF=E-1TOF STEP-8
490 FOR GH=G/50 TO H/50-1
500 LPRINT CHR$(27);CHR$(75);CHR$(50);CHR$(48);
510 FOR HH=1 TO 50
520 OUT 253,PEEK(A)
530 A=A+1
540 NEXT HH
550 NEXT GH
560 LPRINT
570 NEXT EF
580 LPRINT CHR$(27);CHR$(64)
590 PRINT,"1 uj tartomany ertelmezes"
600 PRINT,"2 fugveny beiras"
610 PRINT,"3 nyomtatas"
620 PRINT,"4 VEGE"
630 T$=INKEY$
640 I$=INKEY$:IF I$>"0" AND I$<"5" THEN 650 ELSE 640
650 ON VAL(I$) GOTO 10,260,410,670
660 RESUME NEXT
670 END
680 CLS:PRINT "A fugveny egyenletei: ":HF=17755
690 INPUTZ$,V$:PRINT,"legyel turelemmel":FF=0
700 J$=Z$+":"+V$+":"
710 FOR FB=1 TO LEN(J$)
720 I$=MID$(J$,FB,1)
730 RESTORE
740 FOR AR=0 TO 9
750 READ R$
760 IF I$=R$ THEN  DD=AR+205:GOTO 1030
770 NEXT AR
780 I$=MID$(J$,FB,2)
790 READ R$
800 IF I$=R$ THEN DD=143:FB=FB+1:GOTO 1030
810 I$=MID$(J$,FB,4)
820 READ R$
830 IF I$=R$ THEN DD=202:FB=FB+3:GOTO 1030
840 READ R$
850 IF I$=R$ THEN DD=149:FB=FB+3:GOTO 1030
860 I$=MID$(J$,FB,3)
870 FOR AR=0 TO 13
880 READ R$
890 IF I$=R$ THEN DD=AR+215:FB=FB+2:GOTO 1030
900 NEXT AR
910 READ R$,Q$
920 IF I$=R$ OR I$=Q$ THEN 930 ELSE 1010
930 FOR AR=0 TO 13
940 READ R
950 POKE HF+AR+FF,R
960 NEXT AR
970 IF I$=Q$ THEN POKE HF+FF+5,205
980 FF=FF+14
990 FB=FB+2
1000 GOTO 1040
1010 I$=MID$(J$,FB,1)
1020 DD=ASC(I$)
1030 POKE HF+FF,DD
1040 PRINTI$;
1050 FF=FF+1
1060 NEXT FB
1070 POKE HF+FF+1,147
1080 PRINT:RETURN
1090 DATA+,-,*,/.Ä,&,%,>,=,<,IF,THEN,ELSE,SGN,INT,ABS,FRE,
INP,POS,SQR,RND,LOG,EXP,COS.SIN,TAN,ATN,SHX,CHX
1100 DATA 40,69,33,209,88,206,69,33,209,206,88,41,208,50
```

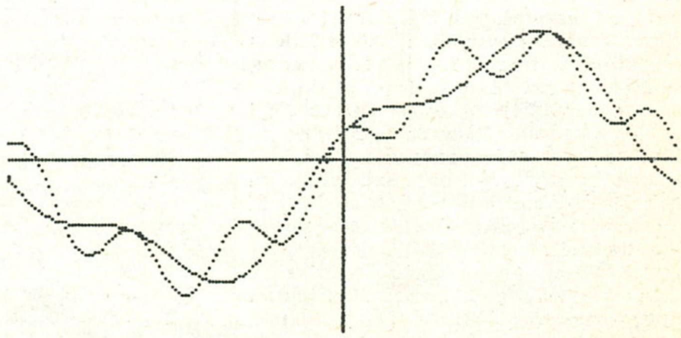
Y=4*SIN(X/P!)+COS(X)::



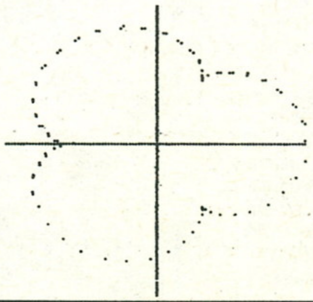
Y=4*SIN(X/P!)::



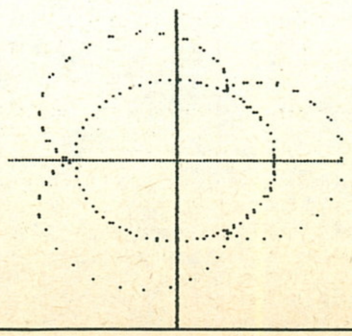
Y=4*SIN(X/P!)+COS(2*X)::



X=4*COS(T)+COS(4*T):Y=4*SIN(T)+SIN(4*T):



Y=3*SIN(T):X=3*COS(T):



A VC-20 és nemcsak a VC-20

Az alábbi írás látszólag csak a VC 20-as gépek nem kis számú híveinek szól. Közlésén gondolkodtunk is, hiszen ahogyan ma már szinte föl sem merül ZX 81-es anyagok közlése, igazából a VC 20-as anyagokért sem vagyunk oda. Figyelmesen végigolvasva a cikket, nyilvánvalóvá vált azonban, hogy a benne lévő információk ugyanúgy érvényesek a C 64-re, a C 16-ra, mint a VC 20-ra. Ami nem egyezik a három gépen, az természetesen a közölt címek egy része. (A BASIC program és a változók tárolásával kapcsolatos memóriacímek azonban azonosak a három gépen!) Így hát természetesen a közlés mellett döntöttünk.

Bár nem vagyok VC 20 tulajdonos, de márkatársként nagyon együttérzek velük, amikor információhiányra panaszkodnak, ezért szeretném közkinccsé tenni a BIT-LET-en keresztül az erről a gépről rendelkezésemre álló adatokat. Röviden a társzervezésről: A BASIC-terület kezdete (innen kezdődik a felhasználói program) lekérdezhető a ?PEEK(43)+PEEK(44)*256 paranccsal. 3 kB-os bővítésnél: 1024. 8, 16, 24 kB-os bővítéseknél: 4608. A skaláris változók kezdőcímét a következő paranccsal kaphatjuk meg: ?PEEK(45)+PEEK(46)* 245. Ezek a változók a program után következnek (tehát a kapott érték az aktuális programhossztól függ) és a program szerinti első előfordulás sorrendjében tárolódnak, először van az azonosító, azután az értéke. Stringváltozó esetén itt csak a string hossza és egy mutató áll, mely a stringmezőben tárolt karaktersorozat kezdőcímét adja. Az értékek tárolási módja: bináris kettes komplemenskód. Itt tárolódnak a felhasználó által definiált függvények is, de szintén csak mutatókkal. A skaláris változók után következnek a tömbök, a kezdőcím: ?PEEK(47)+PEEK(48)* 256 paranccsal iratható ki. A tömbazonosító után a következő tömbre mutató pointer áll, aztán a dimenziószám, majd ezek nagysága következik, s ezek után jönnek a tömb elemei úgy, hogy a dimenzionálás első indexe változik a leggyorsabban (ez kétdimenziós esetben oszlopfolytonos tárolást jelent). A valós elemek 5, az egész elemek 2, a stringek 3 byte-on tárolódnak.
Egy új skaláris változó első alkalmazásakor az egész tömbterület 7 byte-tal magasabb címre kerül, mert a skaláris változók és a tömbök között „helyet kell szorítani" az újonnan bevezetett változónak.
A tömbterület végét megadja a következő parancs: ?PEEK(49)+PEEK(50)* 256. A stringmező kezdete megkapható: ?PEEK(51)+PEEK(52)*256, a vége, mely egyben a BASIC-terület vége: ?PEEK(55)+PEEK(56)*256 paranccsal. Az utóbbira vonatkozó értékek: alapkiépítésben és 3 kB-os bővítésnél: 7680, 8 kB-os bővítésnél: 16384, 16 kB-osnál: 24576, 24 kB-osnál: 32768.
A tömbterület vége és a stringterület kezdete között van a szabad memóriaterület, ennek a nagyságát adja meg a FRE(X) függvény. E terület nem biztos, hogy teljesen üres, tartalmazhat olyan stringeket, amelyekre már nem mutat mutató (mert megváltoztattuk a stringváltozó vagy tömbelem értékét). Ha a program futása során az előbb említett két mező összeér, akkor egy belső eljárás generálódik (GARBAGE COLLECTION), mely megvizsgálja, hogy valamennyi karaktersorozatra mutat-e mutató. Ha igen, akkor OUT OF MEMORY üzenettel leáll a futás, ha nem, akkor a stringeket tömöríti, s ha elegendő a felszabadult hely, akkor folytatódik a program futása. (Vigyázat: az említett hibaüzenetet kapjuk akkor is, ha a verem telt be, például sok ciklust és/vagy szubrutinhívást ágyaztunk egymásba.)
Az ebből levonható következtetések, tanácsok: memóriaigényes programjaink írásakor „bánjunk takarékosan" a stringekkel és a már nem használtakat tegyük üressé. A kezdeti időben

0	Sorvég	66	B	133	INPUT	169	STEP
1-31	Üres	67	C	134	DIM	170	+
32	Space	68	D	135	READ	171	−
33	!	69	E	136	LET	172	*
34	"	70	F	137	GOTO	173	/
35	#	71	G	138	RUN	174	'
36	$	72	H	139	IF	175	AND
37	%	73	I	140	RESTORE	176	OR
38	&	74	J	141	GOSUB	177	>
39	'	75	K	142	RETURN	178	=
40	(	76	L	143	REM	179	<
41	)	77	M	144	STOP	180	SGN
42	*	78	N	145	ON	181	INT
43	+	79	O	146	WAIT	182	ABS
44	,	80	P	147	LOAD	183	USR
45	−	81	Q	148	SAVE	184	FRE
46	.	82	R	149	VERIFY	185	POS
47	/	83	S	150	DEF	186	SQR
48	0	84	T	151	POKE	187	RND
49	1	85	U	152	PRINT#	188	LOG
50	2	86	V	153	PRINT	189	EXP
51	3	87	W	154	CONT	190	COS
52	4	88	X	155	LIST	191	SIN
53	5	89	Y	156	CLR	192	TAN
54	6	90	Z	157	CMD	193	ATN
55	7	91	[	158	SYS	194	PEEK
56	8	92	£	159	OPEN	195	LEN
57	9	93	]	160	CLOSE	196	STR$
58	:	94	↑	161	GET	197	VAL
59	;	95	"	162	NEW	198	ASC
60	<	96-127	Üres	163	TAB(	199	CHR$
61	=	128	END	164	TO	200	LEFT$
62	>	129	FOR	165	FN	201	RIGHT$
63	?	130	NEXT	166	SPC(	202	MID$
64		131	DATA	167	THEN	203-254	Üres
65	A	132	INPUT	168	NOT	255	

tanulságos a ?FRE(X) gyakori használata és a kapott eredmény értékelése.

Visszatérve a társzervezésre, a program tárolását egy ábrán mutatom be. A BASIC utasításkészlet alapszavait egy byte-on tárolja az interpreter, melynek értékét tokennek nevezzük. A mellékelt táblázat a VC 20 tokenjeit tartalmazza decimálisan. A 32 és 95 közötti jelek egyben ASC II-kódok is. A számok tárolása ASC II-kóddal történik.

A fentiek ismeretében olyan fogásokat is alkalmazhatunk, amelyet a VC 20-on alkalmazott BASIC-verzió nem ismer. Néhány ötlet: lokális tömbtörlés. A legutoljára dimenzionált tömböt törölni tudjuk, ha a dimenzionálás előtt elmentjük a tömbterület végét jelző két byte értékét, majd mikor már nincs szükség erre a tömbre, akkor visszatöltjük a korábban megjegyzett értékeket, így helyet szabadíthatunk fel a memóriában. Hasonlóan történhet skaláris változók törlése, de vigyázni kell arra, hogy a felette levő tömböket is mozgatni kell „visszafelé" a tárban, mert ez az interpreter nem teszi meg helyettünk (ő csak „fölfelé" mozgat automatikusan). A végrehajtás a PEEK és a POKE utasítások megfelelő alkalmazásával tehető meg.

Hasznos tudni, hogy a képernyőmemória alapkiépítésnél és 3 kB-os bővítésnél 7680–8191 között, 8, 16, 24 kB-os bővítések esetén 4096–4607 között van, a hozzárendelt színmemória a fenti sorrendben a 37888–38399, ill. 38400–38911 között helyezkedik el.

Gyakorlottabb felhasználás esetén szükség lehet a BASIC terület határainak módosítására. Pl. BASIC-ből gépikódú rutint hívunk vagy karakterkészlet-módosításkor stb. Az elsőnél a gépi kódú program tárolására kell a hely, a másodiknál pl. ékezetes magyar betűk alkalmazásához a ROM-ból a karakterkészletet át kell másolni a RAM-ba, hogy azt módosíthassuk, hiszen általunk csak a RAM írható és utasítanunk kell a gépet, hogy az új készletből dolgozzon.

Az eltolás nagyságát természetesen előre ki kell számolni. Egy példa: A BASIC-terület végét 512 byte-tal lefelé toljuk (7680-tól 7168-ra). POKE 55, 0:POKE 56, 28:CLR és természetesen Return. Magyarázat: 0·1+28·256=7168. A felszabadított 512 byte-on elhelyezhető pl. egy gépi kódú rutin. A BASIC alsó határának felfelé tolása, ill. ezeknek más memórianagyságnál való alkalmazása hasonlóan történhet.

Érdemes még tudni a NEW nem törli a tényleges programot, hanem a skaláris konstansok elejére mutató pointert (45 és 46) állítja át és törli az első sor mutatóját mely a második elejét adja meg.

A továbbiakban néhány, általam ismert POKE utasítást és azok hatását adom közre:

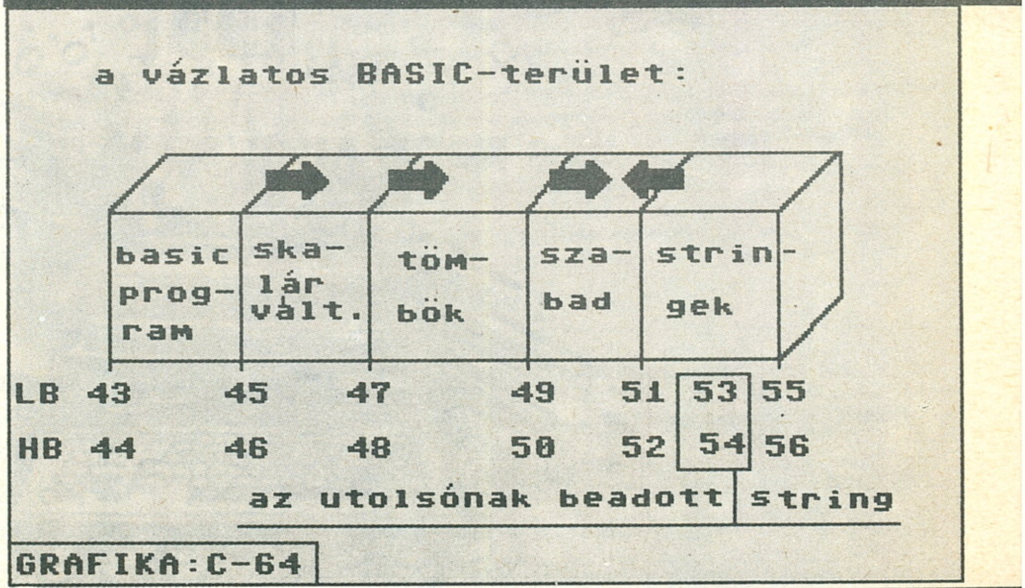
POKE 657, 128 ill. POKE 657, 0	írás kis- és nagybetűvel
POKE 810, 0: POKE 811, 0	letiltja a GET-et
SYS 65499	nullázza a TI és a TIS változókat
SYS 65511	valamennyi file-t lezárja
POKE 199,1	RVS ON
POKE 199,0	RVS OFF
POKE 774,0	sorszámlistázás
POKE 774,200	tiltja a listázást
POKE 792,2	RESTORE hatástalan
POKE 792,173	RESTORE normál
POKE 37148,251	leállítja a magnó motorját

Végezetül a könnyebb érthetőség kedvéért, a cikk első részében leírtakat egy vázlatos ábrával illusztrálom. Az ábrához fűzött megjegyzés: A VC 20-hoz sokféle bővítőmodult használnak. Ezek közül a nagyobbak társzervezése olyan, hogy a bővítésképpen használt munkaterületek több darabban helyezkednek el a memóriában és közöttük más célra használt területek vannak.

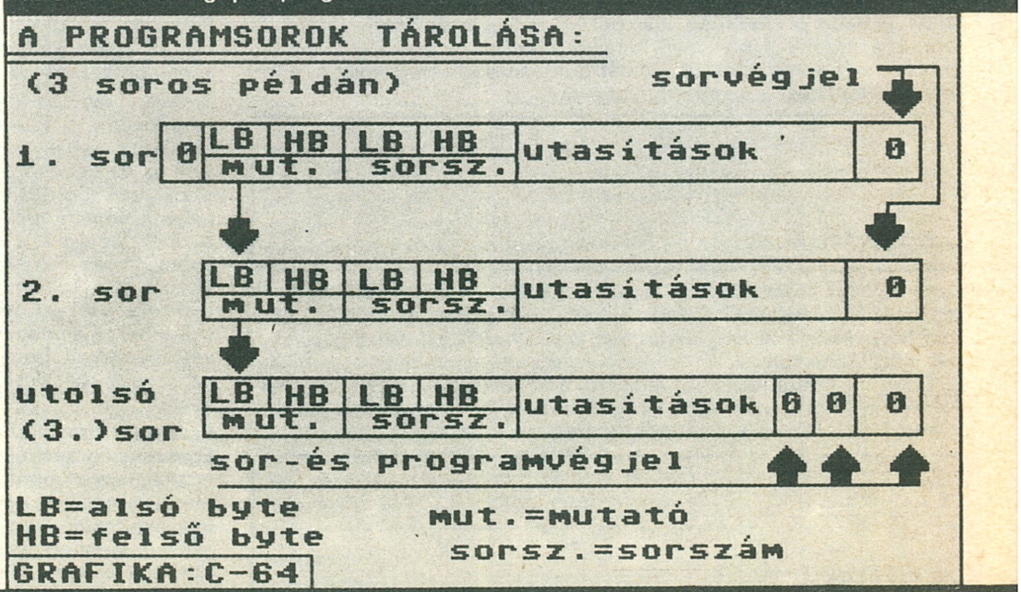
Az alkalmazáshoz sok sikert kíván

Szabó István Budapest, Néphadsereg tér 10/b VI. 2. 1055.

A Commodore-ok társzervezésére



A Commodore-gépek programtárolási rendszere



A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!



Szinte minden példatárban szerepel az alábbi, kezdőknek szóló feladat: Helyezzünk a képernyő soraiba annyi pontot véletlenszerűen, amennyi egy sorba elfér! A feladat pontosítása helyett inkább megadjuk a megoldást – a kedvenc C 16-ra:

```
100 GRAPHIC0,1
110 FOR S=0 TO 23
120 FOR O=0 TO 37
130 CHAR 1,40*RND(0),S,"•"
140 NEXT O
150 NEXT S
190 GETKEY A$
```

Futtatás előtt próbáljuk megtippelni, mennyire lesz „telített" végül a képernyő? Remélem, mindenki kicst meglepődik először, mennyi üres hely maradt, azaz gyakran tesz olyan helyre pontot a gép, ahol már előzőleg is volt.

2. Talán még váratlanabb a hatás, ha a következő változattal próbálkozunk, amikor is nem előre meghatározott az elhelyezendő pontok száma, hanem az első ismétlődésig dolgozik csak a gép:

```
50 DIM A(40)
100 GRAPHIC0,1
110 FOR S=0 TO 23
120 DB=0:FOR O=0 TO 37:A(O)=0:NEXT O
130 O=36*RND(0):IF A(O)>.5 THEN 150
140 DB=DB+1:A(O)=1:CHAR 1,O,S,"•":GO TO  130
150 CHAR 1,36,S,STR$(DB)
160 B=B+DB+1:NEXT S
170 PRINT:PRINT B/24;
190 GETKEY A$
```

„strukturált" változat

```
50 DIM A(40)
100 GRAPHIC0,1
110 FOR S=0 TO 23
120 DB=0:FOR O=0 TO 37:A(O)=0:NEXT O
130 DO WHILE A(O)<1.5
135 O=36*RND(0):DB=DB+1:A(O)=A(O)+1
140 CHAR 1,O,S,"•":LOOP
150 CHAR 1,36,S,STR$(DB)
160 B=B+DB+1:NEXT S
170 PRINT:PRINT B/24;
190 GETKEY A$
```

Praktikus okokból itt már a soronkénti pontszámot is feltüntettük, sőt az egész képernyőre vonatkozó átlagot is kiszámítottuk, mert annyira meglepőnek érezzük.

3. Innen már csak egy lépés a valószínűségszámítás talán legismertebb paradoxona: a születésnap-probléma, ami így hangzik: Mi a valószínűsége annak, hogy egy N fős csoportban van két olyan ember, akik az év ugyanazon napján születtek? Érdemes ezt egy kb. 30 fős osztályban kipróbálni! Mielőtt a P(N) függvényt kiveséznénk, próbáljunk tippelni néhány értékére! Vajon mekkora csoport esetén lesz a valószínűség 1 – könnyű belátni, hogy ***legalább 367*** ember kell ahhoz, hogy teljesen biztosra mehessünk. Az első két feladatból viszont már sejthető, hogy ritkán kell ilyen soká elmennünk. Vajon mekkora eséllyel lesz egy 183 fős társaságban két ember, akik ugyanakkor ünneplik a születésnapjukat? P(N)-nel jelölve annak valószínűségét, hogy egy N létszámú csoportban lesz két ilyen ember, próbáljunk becsléseket adni P(100), P(70), P(50), P(30) értékekre!

Ezek után vizsgáljuk meg alaposabban a P(N) függvényt!

Egyszerűség kedvéért – mint sok más esetben is – az egyes P(N) értékek helyett pont az ellentett esemény valószínűségét számítjuk ki: mekkora eséllyel ***nem lesz*** két ilyen ember, azaz mekkora eséllyel lesz mindenkinek máskor a születésnapja? Ez nyilván 1–P(N) lesz. Állítsuk valahogyan sorba a csoport tagjait (névsor, magasság szerint, ez teljesen lényegtelen a továbbiakban), és nézzük az első személyt. Ő születhet bármikor, szökőévet is figyelembe véve az év 366 napjának bármelyikén. A második emberrel már nem lehetünk ilyen nagyvonalúak, neki már „csak" 365 nap jut, a harmadiknak 364..., az n-ediknek már csak 366—n+1=367—n. Felhasználva a független események szorzására vonatkozó összefüggést, és kedvező összes lehetőségek arányát, könnyen adódik, hogy

$$1-P(N)=\frac{366}{365}\cdot\frac{365}{366}\cdot\frac{364}{366}\cdots\frac{(367-N)}{366}$$

A formula ideálisan programozható, így gyorsan ki is írathatunk – megfelelő elképedés közepette – néhány értéket.

1. változat

```
50 P=1:NAP=366
100 FOR I=0 TO 11
110 FOR J=1 TO 10
120 P=P*(NA+1-(I*10+J))/NA
125 VE=VE+P
130 NEXT J
140 PRINT (I+1)*10;TAB(5);"F0";TAB(12);1-P
150 NEXT I
160 PRINT "ELS0 ISM. VHATO E.=";VE
```

2. változat

```
30 DIM P(120)
50 P=1:NAP=366
100 FOR I=1 TO 120
120 P=P*(NA+1-I)/NA:P(I)=1-P
130 NEXT I
150 INPUT N:IF N<2 THEN END :ELSE PRINT N,P(N):GO TO 150
```

Ismert, hogy a V valószínűségi változó várható értéke a

$$\sum iP_i = \sum i \cdot P\{V=i\} \text{ képleten kívül } \sum P\{V>i\}.$$

formában is kiszámítható. Ezt használtuk ki az 1. változatban.

4. Teljesség kedvéért nézzük meg a függvény görbéjét is. Célszerű többszínű grafikát használni, mert P(N) már N>100 esetén is „érdektelen", viszont a „nagyfelbontás" vonalai túl vékonyak!

```
50 P=1:NAP=366
100 GRAPHIC 3,1
110 FOR I=1 TO 155
120 P=P*(NA+1-I)/NA:PP=190-188*(1-P)
130 IF I/10=INT(I/10) THEN DRAW 1,I,PP:ELSE DRAW 1,I,190 TO I,PP
150 NEXT I
```

5. Érzem annyira meglepőnek az eddigieket, hogy ne álljunk meg itt. Hitetlenkedők kedvéért „szimulálni" fogjuk ezt a szituációt. Aki a szó hozzávetőleges jelentését sem ismeri, annak Király Zoltán írását ajánljuk (BIT-LET 1985. június 27. – LIFT). Az alábbi program véletlen születésnapokat generál, egészen az első egyezésig. Az egyezést nem tömbbel figyeli (mint II. és III. programokban), hanem a memóriatakarékosabb PEEK függvénnyel a 150-es sorban. 87 az üres kör O képernyő-kódja. A 145-ös sorban azért kell módosítani az I, J értékeket, hogy maradjon hely a hónapok neveinek, és a napok számainak.

```
10 DATA 0,31,60,91,121,152,182,213,244,274,305,335
12 DATA JAN,FEB,MAR,APR,MAJ,JUN,JUL,AUG,SZE,OKT,NOV,DEC
30 DIM H(12)
40 GRAPHIC 0,1
50 FOR I=1 TO 12:READ H(I):NEXT
70 CHAR 1,5,2,"1234567890123456789012345678901"
72 CHAR 1,5,1,"         1111111111222222222233"
80 FOR I=4 TO 15:READ A$:CHAR 1,1,I,A$:NEXT
110 A=INT(367*RND(0)+1):N=N+1
120 FOR I=1 TO 12:IF A>H(I) THEN 140
130 I1=I:I=15
140 NEXT I
142 IF I<14 THEN I1=I
145 I=I1-1:J=A-H(I)+5:I=I+3:P=3072+40*I+J
150 IF PEEK(P)<>87 THEN CHAR 1,J,I,"o":GO TO 110
160 CHAR 1,J,I,"●":CHAR1,2,20,""
170 PRINT N;".-IK SZEMELY MAR ISMETLES!"
```

Ha valakit még ez sem győz meg, és lusta osztályban, baráti körben kísérletezni, akkor ... J. G. Kemény (bizony, a BASIC egyik atyja) – J. L. Snell–G. L. Thompson könyvében „A modern matematika alapjai"-ban olvasható a következő (Műszaki Kiadó, 1871. 154–157. oldalak): Az Egyesült Államoknak 1970-ig 37 elnöke volt. Pork és Harding elnökök november 2-án születtek. Az elnökök közül 1970-ig 35-en haltak meg. Jefferson, Adams és Monroe július 4-én haltak meg!

6. Kicsit rímel az eddigiekre a következő feladat: mi a valószínűsége annak, hogy él Budapesten két olyan ember, akiknek ugyanannyi szál haja van? Igen könnyű belátni, hogy biztosan van két ilyen személy – még ha a teljesen kopaszoktól el is tekintünk, akkor is! Ehhez csak azt kell tudni, hogy bárkinek félmilliónál kevesebb szál haja van. A harmadik program minimális változtatásával „megvallathatjuk" ezt a függvényt is. (A 110, 120, 130-as sorokban 10 helyére tetszőleges azonosítót írunk!) Bizony meglepő, hogy egy nagyobb iskolában (900 fő körül) már ötven százaléknál nagyobb az esély két „egyforma fejű" gyerekre! Egy közepes faluban pedig (3000 lakos), már ***egy ezreléknél jóval kisebb arányban*** várható csupa különbözően „borzas" személy! Az első ismétlődés várható értéke – félmillióval alaposan túlbecsülve az emberi hajállományt! – durván 885 fő! Kettőszázezer hajszálat feltételezve pedig, mindössze 500 személy a várható érték. Ezen a ponton a probléma már nem is paradoxon, hanem a számítógéptudomány egy igen érdekes fejezetecskéje: a „hesselés" (angolul hashing) problémája, amivel részletesen foglalkozik D. E. Knuth klasszikus könyve: The Art of Computer Programming 3. kötet. A könyv magyar fordítása hamarosan várható a Műszaki Kiadónál.

7. Csak rövid ízelítőt kívántunk nyújtani a valószínűségszámítási érdekesség világából. További furcsaságok olvashatók. W. Weaver: Szerencse kisasszony (Gondolat, 1979) könyvében igen érdekes, anekdotázó stílusban. Igényesebbeknek Székely J. Gábor: Paradoxonok a véletlen matematikájában (Műszaki Kiadó, 1982) művét ajánlanám. A születésnaphoz hasonlóan egyszerű problémáktól igazán bonyolultakig ível, és közben sok történeti érdekességgel is szolgál, elvégre „a legtöbb tudomány története – a matematikáé is – paradoxonok története".

Török Turul

PROGRAM BÖRZE

Szeretne hatékonyabban programozni C-64-est?
Jobban kihasználni grafikus printerének adottságait? Ebben nyújt segítséget Önnek a PRINTER BASIC nyelv! PRINTER BASIC: Mi ez?
A C 64 alap BASIC nyelvének lényeges bővítése. Új, kezdők által is kényelmesen használható BASIC utasításokat szolgáltat a grafikus nyomtatók kezeléséhez. A nyomtatón létrehozható ábrák méretei:
Vízszintesen 480 pont
Függőlegesen 392 pont
A PRINTER BASIC 8 Kbyte-os cartridge ROM-ban kerül forgalomba, amelyet a C 64-es expanziós pontjába kell helyezni. A gép bekapcsolása után a PRINTER BASIC rövid üzenettel azonnal jelentkezik.
A cartridge ára: 6900 Ft
Iskolák részére: 3600 Ft
Megrendelésre elküldjük Önnek – tesztelésre is – a PRINTER BASIC cartridge-t a felhasználói kézikönyvvel együtt. Ha megnyeri tetszését, megvásárolhatja, amennyiben nem találja szükségesnek munkájához, úgy a küldemény visszajuttatása érvényteleníti megrendelését!
Címünk: HARDSOFT Számítástechnikai GM
2440 Százhalombatta, Rózsa F. u. 32.

Eladom vagy elcserélem **Commodore 16, ill. 64 számítógépre készített angol és orosz nyelvi oktatóprogramjaimat.**
Tel.: 143-031, 330-345 Kálmán Albert

C-64 gépi kódot tanulók!
Elemzésre felajánlok két gépi kódú programot (kétjátékosos lövöldözős, illetve teljes képernyős életjáték). Visszafejtésük hasznos lehet a 6510-es utasítások használatának megértéséhez.
Akit érdekel, küldjön kazettát, és megcímzett, felbélyegzett válaszborítékot. Érdeklődni lehet az 558-958-as telefonon, vagy levélben.
Nagy Bálint, Bp. 1122 Gaál József u. 4.

Commodore 64 **gépre angol–I. (3 lemez), angol–II. (4 lemez), német–I. (3 lemez), német–II. (4 lemez) teljes tanfolyami anyagot tartalmazó programok kaphatók lemezenként is.**
ADFELOR GMK. 1027 Budapest, Mártirok útja 26. fsz. 3.
Tel: 151-224

Ajánlom C-64-re írt programomat, **amely a mezőgazdasági nagyüzemeket érdekelheti elsősorban. Két alprogramból áll, amely külön-külön is futtatható. Az első rész a szántóföldi növényvédőgépek (permetező, porozó és granulátumszórók) beállítását segíti. Az ehhez szükséges számításokat elvégzi a felhasználó igényei szerint, de ajánlatot tesz a beadott adatok alapján a legkédvezőbb beállításra (idő- és üzemanyagtakarékosság).**
A programlemezen tárolja az érvényben lévő növényvédőszerek engedélyokiratának legfontosabb előírásait, így a szer nevének megadása után kinyomtatja a munkavédelmi, elsősegélynyújtási és környezetvédelmi tudnivalókat.
A program végeredményként – a megadott adatok alapján – minden tekintetben teljes Növényvédelmi Technológiai Munkautasítványt készít.
Érdeklődés esetén a helyszínen bemutatom.
Lázár József, Szentes, József A. u. 8/a. 6600

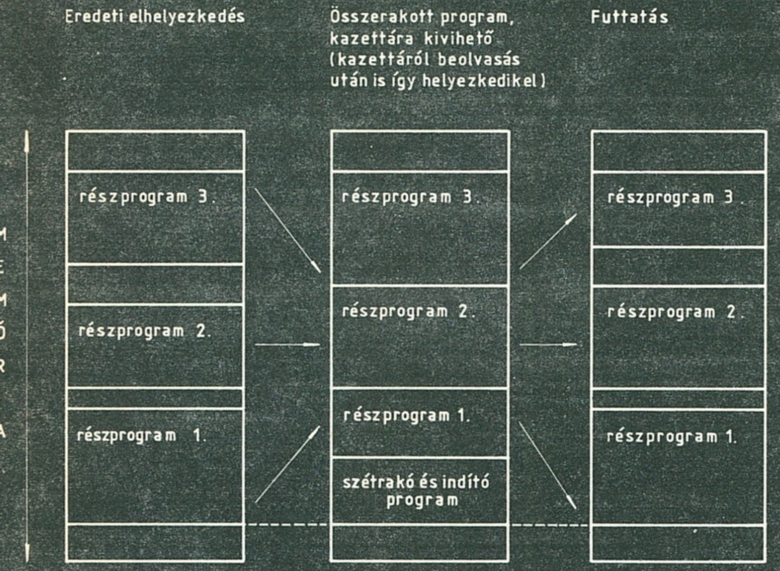
Áruházak, reklámügynökségek részére ajánljuk reklámszövegszerkesztő programjainkat! **Könnyen, szaktudás nélkül állíthat össze Commodore 64 számítógéppel és VC 1541 floppyval reklámszövegeket és grafikákat tartalmazó műsorokat, melyeket a kirakatban folyamatosan bemutathat!**
Videóra is felvehető!
Kérjen díjmentes tájékoztatót!
AJKASOFT SZÁMÍTÁSTECHNIKAI GM., Ajka, Vöröshadsereg út. 110.

POSTA

BP. 1986

Tisztelt Szerkesztőség!
Rengeteg program, köztük sok játékprogram csak floppyról jön be a C 64-es gépen. Sok társam próbálkozik ezeknek magnóra írására, több-kevesebb sikerrel. Azokat a programokat, amelyek több részből állnak, hogyan lehet átírni magnós változatra?
Van-e olyan utasítás, ami a gépi kódú programot kilistázza, hogy „bele lehessen nyúlni''?
Van ezzel kapcsolatban egy másik problémám is. A TURBO TAPE gyorsítóprogram azt hiszem, olyan helyre töltődik, ahová a betöltendő főprogram kerül. Így van ez? Ha igen, hogyan lehetne ezt kiküszöbölni, hiszen emiatt igen sok jó program hibás lesz, nem lehet futtatni.
***Bognár Ákos**, Budapest, Levendula u. 20. 1124*

A többrészes programok átírására a legjobb (és legelterjedtebb) módszer szerint először is egyrészest kell belőle csinálni. Az eljárás a következő:
A részprogramokat nem az eredeti helyükre kell tölteni, hanem közvetlenül egymás után, majd ki kell egészíteni egy rövid gépi kódú rutinnal, ami indításkor a programrészeket a helyükre teszi, és az eredeti program indítási címére adja a vezérlést.



Két dologra kell ügyelni:
1. A részprogramok mozgatásának sorrendjére ügyelni kell, nehogy fontos programrészeket átírjon.
2. Szalagról történő betöltés esetén, ha az a BASIC vagy KERNAL ROM alá kerül, ? LOAD ERROR hibajelzést kapunk. (Ekkor nem a szalag rossz, sőt a program is jól töltődik be.) Tudomásunk szerint a V 3–TURBO az egyetlen turbo program, amely ezt a fölösleges hibajelzést kivédi.
Kétfajta program esetén a szalagra másolása a fenti módon nem oldható meg:
– amelyik futás közben aktívan használja a floppyt; adatokat, programrészeket tölt be róla,
– amelyik betöltés után vagy közben azonnal indul. Ezek egy része kazettáról töltve nem működik, ezért a betöltő programrészt át kell írni. Ehhez viszont a gépi kódú programozás alapos ismerete szükséges.
Gondot jelenthet még, hogy a betöltendő program töltés közben átírja a TURBO TAPE programot. Ez ellen megfelelő védelem a program egyrészessé tétele. Sok turbo elszáll hosszú, bár egyrészes programok betöltésekor is. Ilyen esetben a V 3–TURBO használatát javasoljuk.
A bővítetlen gépen gépi kódú programok kilistázására nincs utasítás. Az ún. DIASSEMBLER programokkal lehet ezt (is) megtenni. Ezek közül a legelterjedtebbek:
– PROFI MON 64
– SUPERMON
– HESMON
– HELT-PLUS

Chrabák Ede

Önök elég sokat foglalkoztak az ÁPISZ akciójával (akkor vettem én is a gépet). Azt hiszem az ÁPISZ akkor nem állt le. Tegnap kaptam egy levelet, melyben tájékoztatnak arról, hogy új játékkazetták érkeztek, valamint előjegyzést vesznek fel memóriabővítőre és botkormányra. És az ár az általam ismertek között a legolcsóbb! Azt hiszem itt kezdődik az igazi kereskedelem. Remélem Önök továbbra is közölnek C 16 programokat. Ha lehet játékprogramot is, mert jelenleg sajnos nekem nincs. Valamint szeretném kérdezni van-e lehetőség egyes C 64-es programokat, melyek POKE utasításokat is tartalmaznak, C 16-ra átírni. Válaszukat köszönöm.
***Demeter Árpád**, Szolnok, Várkonyi tér 20. IV. 5. 5000*

Kérdésére válaszolva:
1. Játékprogramot nem túl gyakran, de közlünk.
2. Igen, a C 64-es programokat megpróbálhatja átírni. Persze tudni kell, hogy minek mi a megfelelője. (Vannak POKE-k, a C 64-en, amelyek például a 16-oson egyszerű BASIC utasításokkal helyettesíthetők.)
Ami nem megy a 16-oson, az a sprite-kezelés, márpedig komolyabb 64-es programok – például a játékok – ezt nem nélkülözhetik.

Két kérdésem van:
1. Igaz-e, hogy a Commodore 64-re írt „AUSTRO COMP'' program csak lemezegységgel használható?
2. Ha igen, akkor van-e ehhez hasonló program magnóra?
***Kiss Róbert**, Körösladány, Dózsa Gy. 78. 5516*
Igen, igaz. Sajnos nem tudunk hasonló kazettáról betölthető programról.

Lehetőségem lenne egy Sinclair Spectrum számítógép megvételére. Szeretném megtudni, hogy a ZX Spectrum 48 K-s játékprogramok futtathatók-e Sinclair Spectrum + gépen? Lehet-e csatlakoztatni ZX Spectrum interface-eit Sinclair Spectrum+ géphez?
***Szabó Zoltán** Budapest, Marx u. 105. 1181*
*Ugyanez a kérdése **Antalics bélának** (Bp. Pusztaszeri út 9/B. 1025) is.*

Lehet, hogy korábbi írásainkból nem derült ki egyértelműen: a Spectrum és a Spectrum+ gépek kizárólag a billentyűzetben térnek el, egyébként teljesen megegyeznek, így a programok futtathatók és az interface-ek is csatlakoztathatók.

Nekem is az a szerencsés problémám adódott, hogy hamarosan döntenem kell, milyen számítógépet vásároljak. Kérem segítsenek a választásban. Elsősorban a C 64-es és a C 128-as gép összehasonlítására lennék kíváncsi a hang, a képernyőkezelés és az esetleges szoftverkompatibilitás szempontjából. Azt is szeretném tudni, hogy a C 128-as kompatibilis-e a 64-eshez használható perifériákkal.
***Szabó Ferenc**, Szentlászló, Zrínyi u. 3. 8893.*

1. A C 128-asnak két „üzemmódja'' van. Az egyikben teljes egészében azonos a gép a C 64-essel, tehát mindent tud, amit a 64-es. A 128-as üzemmódban a C 64 BASIC szinten alulról kompatibilis a C 128-as géppel. Minden „tiszta'' BASIC-ben megírt (lefordítás nélkül) program változtatás nélkül futtatható C 128-as üzemmódban.
A két gép memória kiosztása nem egyezik meg. (POKE, PEEK, SYS címzések nem kompatibilisek.)
2. Hang:
A C 128-as hangjának programozására a
SOUND
ENVELOPE
VOL
TEMPO
PLAY és
FILTER
utasítások állnak rendelkezésre.
3. Képernyőkezelés:
Érvényesek az 1-es pontban leírtak. A C 128-as több utasítással bővített BASIC-je az ún. ablakos módszer használatát is lehetővé teszi (WINDOW utasítás).
4. Szoftver kompatibilitás:
A gépi kódban megírt és a lefordított programok nem futtathatók a C 128-as üzemmódban.
5. Perifériák:
A szabványos MPS nyomtatók ugyanúgy használhatók, mint a C–64 esetében. A nyomtatás programozását a C 128 BASIC 7.0 verziója jobban támogatja.
A C 64-es géppel elkészített file szerkezet a C 128-as üzemmódban is használható változtatás nélkül.

Kisfelbontású GRAFIKA

A Commodore 16 számítógépek legnagyobb előnye, ahogy ez a Vallatóból is kitűnt a kiváló programozási nyelv. A CBM 3.5 BASIC biztosítja számunkra, hogy a gép szinte valamennyi hardver lehetőségét elérhessük „emberközeli", egyszerű utasítások segítségével. Ennek köszönhetően a C 16-tal foglalkozó – amúgy is szűkös – irodalom inkább a BASIC lehetőségeit boncolgatja, a gépi kódú programozásról a fontosabb memóriacímekről kevés információ jut el a felhasználókhoz.

Az alábbi programozási „fogások" ismertetésével főleg azok számára nyújthatok segítséget, akik a nagyfelbontású grafika alkalmazása helyett (amely 16 K-s kiépítésben alig hagy helyet a program számára) inkább a karakteres üzemmód segítségével próbálják színesebbé tenni a programjaikat.

1. Programozható színbillentyűk

A képernyőre kerülő karakterek színét BASIC-ből kétféleképpen határozhatjuk meg: a COLOR 1, ⟨szinkód⟩, ⟨fényerő⟩ utasítással, vagy a PRINT utasítással, az idézőjelen belül elhelyezett vezérlőkarakterek segítségével. Ez utóbbi kényelmesebb megoldás. Hátránya, hogy így csak 16 színt használhatunk a 121-ből. Ezt a 16 színt mi is tetszés szerint kiválaszthatjuk. Az egyes billentyűkhöz tartozó színkódokat ugyanis a gép két helyen tárolja a memóriában. A RAM-ban a 275–290 ($0113–$0122) a ROM-ban pedig (az alapbeállítást) az 57667–57682 ($E143–$E152) címeken. Az, hogy melyik táblázatban kódolt színek kerülnek kijelzésre, a 2041 ($07F9) cím 7. bitjének állapota határozza meg (0–7 bitek=8 bit). Ha a bit magas, a ROM-ban, ha alacsony (bekapcsolás után ez a helyzet), a RAM-ban lévő színtabella „él".

A táblázat a színkódokat úgy tárolja, mint a képernyő-színmemória, vagyis a 0–3. bitek határozzák meg a színt, a 4–6. bitek a fényerőt, a 7. bit pedig a FLASH-flag. Így a kurzor ütemében villogó karaktereket is előállíthatunk egyetlen vezérlőkarakter segítségével.

A vezérlőkarakterek jelentése így megváltozik a C 64-en megszokotthoz képest. Így az inverz E betű nem a fehér színt jelenti, hanem a ⟨CTRL⟩2 billentyűhöz rendelt színt. (Bekapcsolás után ez természetesen fehér.)

A mintaprogram 1. része megváltoztatja a színtabellát, az eredeti színek helyett a C 64 színeit rendeli a megfelelő billentyűkhöz. Az ⟨F1⟩ gomb lenyomása után a C 64 színeit használhatjuk, az ⟨F2⟩ gombbal pedig visszatérhetünk a ROM-ban tárolt, eredeti színekhez.

2. Többszínű (MULTICOLOR) karakterek

A C 64 számítógéphez hasonlóan a C 16 is lehetővé teszi több színű karakterek megjelenítését. A többszín üzemmódot a 65287 ($FF07) cím 4. bitjének magasra állításával kapcsolhatjuk be:

POKE 65287, PEEK (65287) OR 16

Ekkor az inverz karakterek helyett is normál karakterek jelennek meg és eltűnik a kurzor. Ebben az üzemmódban a ⟨CTRL⟩ 1 – ⟨CTRL⟩ 8 billentyűkkel az adott (és a fentiekben ismertetett módon programozható) színű **normál** karaktereket kapjuk, a ⟨COMMODORE⟩ 1 – ⟨COMMODORE⟩ 8 billentyűkkel pedig **többszínű** karaktereket melyeknek első színe a ⟨CTRL⟩ (nem a ⟨COMMODORE⟩) lenyomásával előállítható szín, a 2. és 3. szín pedig a 65303 ($FF17) és 65304 ($FF18) címeken tárolt segédszínek.

Egyszerűen, a PRINT utasítások után elhelyezett vezérlőkarakterekkel szabályozhatjuk, hogy normál, vagy többszínű karakterek jelenjenek meg.

Épp úgy, mint a több színű bit-térképes üzemmód bekapcsolása esetén, a több színű karakterek vízszintes felbontása is felére csökken, mivel így 2 bit határoz meg egy pontot. (00=háttérszín; 01=multi 1; 10=multi 2; 11=multi 3).

Természetesen saját több színű karaktereket is definiálhatunk a normál karakterekhez hasonlóan. (Ld. a BIT-LET által közölt karakter tervező programot, valamint dr. Ury László Commodore 64 c. könyvének II. kötetét.)

A multi color üzemmód az alábbi utasítással szüntethető meg:

POKE 65287, PEEK (65287) AND 239

A minta program 2. része a több színű karakterek használatát mutatja be:

2. A két karakterkészlet egyidejű megjelenítése

A C 16 lehetőséget nyújt arra, hogy egyidejűleg mindkét, a nagy betű/grafikus és a kis betű/nagy betű karakterkészletet megjeleníthsük a képernyőn. A második karakterkészlet az inverz karakterek „helyére" kerül, így azokat nem használhatjuk.

Ezt az üzemmódot a 65287 ($FF07) TED-regiszter 7. bitjének magasra állításával kapcsolhatjuk be:

POKE 65287, PEEK (65287) OR 128

Ezt követően a PRINT utasítások mögött a ⟨RVS ON⟩ és ⟨RVS OFF⟩ vezérlőkarakterek közötti szövegrészek nem inverzben a kis betű/nagy betű karakterkészlet megfelelő betűivel kerülnek kijelzésre.

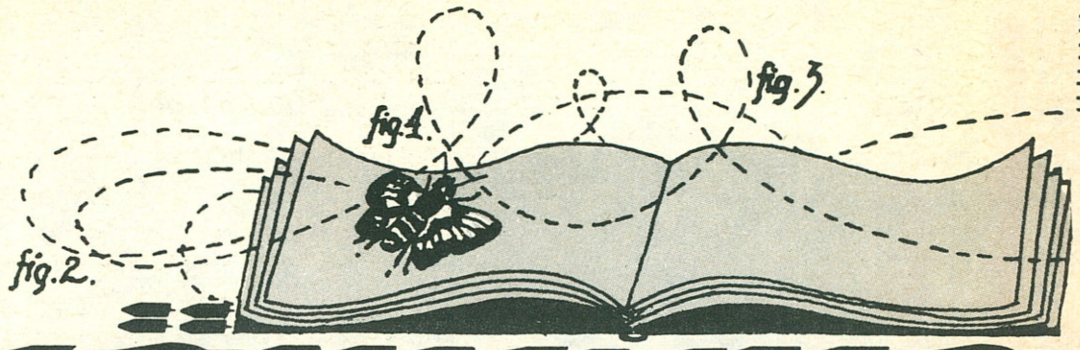
Megváltozik a képernyőkódok jelentése is; a 127-nél nagyobb kód a második karakterkészlet elemét jelöli, így a 129 képernyőkód az inverz A helyett kis a betűt eredményez.

Kilépés az üzemmódból:

POKE 65287, PEEK (65287) AND 127

Morvai László

```
10 KEY1,"POKE 2041,0"  +CHR$(13)
15 KEY2,"POKE 2041,128"+CHR$(13)
20 :
25 POKE 285,107:  REM C= 3 ROZSASZIN
30 POKE 286,49 :  REM C= 4 SZURKE 1
35 POKE 287,65 :  REM C= 5 SZURKE 2
40 POKE 288,95 :  REM C= 6 VIL. ZOLD
45 POKE 289,109:  REM C= 7 VIL. KEK
50 POKE 290,97 :  REM C= 8 SZURKE 3
55 :
60 END
70 :
80 :
90 :
110 CIM=65287:     REM $FF07
115 M1 =65303:     REM $FF17
120 M2 =65304:     REM $FF18
125 :
130 POKE CIM, PEEK(CIM) OR 16
135 REM  MULTICOLOR BE
140 POKE M1,69:    REM MULTI1 = ZOLD
145 POKE M2,65:    REM MULTI2 = SZURKE
150 :
155 PRINT "■NORMAL":       REM <CTRL>+3
160 PRINT "▒MULTICOLOR": REM <C=>  +3
165 :
170 END
```



KÖNYVMO

Seres–Fenyő–Balogh: **A FORTH programozási nyelv.**
Műszaki Könyvkiadó, 291. o. 84 Ft
(A FORTH – a BASIC mellett – a személyi számítógépek leggyakrabban használt nyelve, mivel igen gyors programfutást tesz lehetővé, programokban is jól használható nyelv jellemzőit mutatja be, a FORTH-ban való gondolkodástól a kompilerek jellegzetességeinek tárgyalásáig.)

Plenge–Szczepanowsky: **SIMON's BASIC gyakorlatok**
Data Becker–Novotrade, 225 o. 355 Ft
(Részletes magyarázatokkal és ellenőrző kérdésekkel ellátott „tankönyv", mely az eredeti SIMON's BASIC kézikönyvnél érthetőbben és pontosabban ismerteti a BASIC-bővítés utasításait, kiemelve annak gazdag grafikai és zenei lehetőségeit.)

Lother: **Gépi kódú programozás haladóknak – C 64, PC 128**
Data Becker–Novotrade, 122 o. 319 Ft.
(A könyv – mely a „Gépi kódú programozás a C 64-esen" c. kötet folytatásának tekinthető – a magas szintű gépi kódú program elkészítéséig vezeti el az olvasót. A programozásban jártasak számára is hasznos ismereteket nyújt a megszakítástechnikának és az operációs rendszer rutinjainak kimerítő elemzése.)

Erdős Iván: **IBM PC, XT információs kártya**
LSI ATSZ 72 o. 135 Ft
(Az IBM PC/XT felhasználói számára szükséges információk, táblázatok áttekinthető gyűjteménye)

Dr. Ferenczy Antal: **C 64 Start** (BASIC tankönyv)
LSI ATSZ 167 o. 170 Ft
(A tankönyv – elméleti bevezető után – mintaprogramok segítségével mutatja be a C 64 kezelését, utasításait. A gép BASIC nyelvét nyelvleckeszerű szakaszokból sajátíthatja el az olvasó.)

Dr. Dobay Péter: **Mikroszámítógépes programkatalógus**
LSTI ATSZ 178 o. 184 Ft
(Az évente megjelenő katalógus legfrissebb kötete az IBM és a Commodore gépcsaládra kidolgozott programok leírását tartalmazza, témakörök szerinti csoportosításban.)

Mikroszámítógép kiállítások tapasztalatai
LSI ATSZ 273 o. 228 Ft
(A kötet az 1985. évi Hannoveri Vásár, a párizsi SICOB és a müncheni PRODUCTRONICA kiállítás tapasztalatait összegzi, bemutatva az eszközök, alkatrészek, a készülékek, rendszerek, valamint az alkalmazás legkiemelkedőbb újdonságait.)

Pál–Révbíró: **Hetedhét Commodore Plus 4**
Novotrade, 152 o. 99 Ft
(A C 16 gépet – melynek programjai futtathatók a Plus 4-en – bemutató „Hetedhét" kötet anyaga, a Plus 4 beépített szoftvereit ismertető kiegészítéssel.)

F. Da Costa: A kalandprogram írásának rejtelmei. Hogyan írjunk BASIC nyelven az iskolaszámítógépre kalandprogramot? Műszaki Könyvkiadó, 277 o. 63 Ft

A megyei időszámítás 1341. évének egy áprilisi reggelén történt, hogy a köztiszteletben álló Bilbó Baggins úrhoz – aki a hobbitok népéből való – beállított Szürke Gandalf, a mágus. Vele jött tizenhárom törpe is: Tölgyfapajzsos Thorin, a száműzött királyi utód és kísérete. Nem sok idő telt bele, egy kiadós lakoma és beszélgetés után Bilbó, maga sem értve, miért teszi, felkerekedett velük, hogy messze Keleten, a suhatagbeli Erebor alatt felkutassák a nagy törpekincset, a királyok kincsét.
Hosszú útjuk során számtalan híres tettet hajtanak végre, ismeretlen népekkel keverednek harcba és barátságba. A képernyőn sorra tűnik fel Középföld, Erdőelve, Ködhegység, Vadonföld – a kalandos utazás állomásai. A képek alatt pedig az új és új veszedelmek leírása olvasható, és a számítógép előtt ülő Bilbónak (vagy a játékosnak, aki a kis hobbit viszontagságait éli újra) döntenie kell, hogy hogyan vágja ki magát.
A számítógépes játék címe: The Hobbit, mely Tolkien azonos (magyarul: A Babók) című regénye alapján készült. Szöveges játék: a játékosnak be kell billentyűznie a kalandos helyzetekben helyesnek ítélt utasításokat; a kísérő grafika csak a színhelyeket illusztrálja. A géppel való kommunikáció nyelve a külön e célra kidolgozott, leegyszerűsített angol, az Inglish/Inglish.
Az ehhez hasonló játékprogramok, melyekben a gépbe bevitt, illetve az általa kiírt szövegeké a fő szerep, a számítástechnika hőskorában, a nagyszámítógépek idején voltak egyeduralkodóak, amikor a gépek még nem rendelkeztek a ma jellemző grafikai lehetőségekkel. Napjainkban is készülnek ilyen játékok – vagy azért, mert irodalmi alapanyagra épülnek, vagy mert a gépnek és a játékosnak olyan információkat kell egymással közölniük, melyek nehezen volnának kifejezhetőek grafikával. Sok szempontból érdekesebbek is ezek a játékok, mint a vizuálisan kiválóan megjelenített űrháborúk és üldözések.
F. Da Costa könyve, „A kalandprogramírás rejtelmei" fő részében egy ilyen, szöveges játék (Kardhalak és Kincsek) elkészítésének módját írja le, TRS–80 – vagy ami ennek megfelel: HT 1080Z – számítógépre. Részletes segítséget nyújt a program elkészítéséhez kedvet érző olvasónak, a játék alapötletétől a program alkotóelemeinek, rutinjainak kidolgozásán át a teljes programlista bemutatásáig, elemzéséig. Áttanulmányozása után bárki, aki valamelyest is konyít a HT gép programozásához, képessé válik hasonló izgalmas kalandjáték megírására.
A kötet második része ezeket az ismereteket bővíti ki egy grafikai elemeket is tartalmazó játék (Szörnyek az útvesztőben) kidolgozásával. A szerző a végsőkig kiaknázza a TRS, illetve a HT gépek nem túl gazdag grafikus lehetőségeit.

Nemcsak azoknak ajánlható e könyv, akik saját készítésű játékprogramokkal akarják agyonütni üres óráikat. Ki ne járt volna még úgy, hogy belefogott egy program írásába, de félbehagyta – mert túl bonyolulttá és áttekinthetetlenné vált a programlista, mert lefutása kivárhatatlanul lassú volt, vagy mert a gép memóriája kevésnek bizonyult a bevitt adatok tárolására? A kötet számos ötletes fogást mutat be a programok struktúrálására, az adatok tömörítésére, rendezésére, a gyors adatelérésre. Több fejezet foglalkozik a programfutás közben beadott, nem BASIC-nyelvű parancsok értelmezésével. Az itt leírtak jól hasznosíthatóak olyan programok írásánál, melyek laikus felhasználók számára készülnek. Külön kiemelésre méltó a fordító, Bán Péter munkája, aki az angol és a magyar nyelv eltérő sajátosságaiból, nyelvtanából adódó nehézségeket kiválóan áthidalta e parancsoknál.

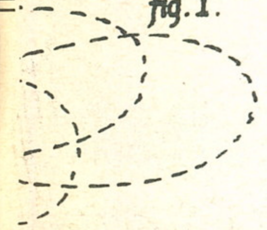
Bilbó volt az első hobbit, aki híressé vált, de bizonyára nem az utolsó, – a kötet elolvasása után számos hobbit, vándor és kalandor indulhat útjára, hogy izgalmas játékban (és tömör, gyors programban) legyőzze ellenségeit.

Plenge–Szczepanowsky: SIMON's BASIC – Gyakorlatok Data-Becker – Novotrade, 225. o. 355 Ft

A Commodore 64-es gép a számítástechnika bogárhátú Volkswagenje.

A számítógépek elterjedésének korai időszakában ugyan sokan más típusú gépeken tanulták a programozás alapjait – Magyarországon ez főképp az ABC 80 volt – de ez csak egy szűkebb, abban az időben vájtfülűnek számító réteget érintett. Az első, igazán tömegesen elterjedt, mindenki által ismert

fig. 1.



LY

mikrogép kétségtelenül a C 64-es, az 1984. év számítógépe. Kiváló játékgép, jók a grafikus lehetőségei, így számos háztartásban megtalálható. Sokak számára egyenlőségjel tehető a „C 64 és a „számítógép" fogalmak közé. Népszerűségét elsősorban alacsony ára és a korábban példátlan szoftverválaszték segítette elő.

Ennek „köszönhetően" olyan feladatokhoz is igénybe veszik (igénybe próbálják venni), melyek ellátására valójában nem alkalmas. Számos vállalat irodáiban megtalálható, ahol pénzügyi, nyilvántartási munkák egyszerűsítésére vásárolták, – itt többnyire kudarcot vallott.

Sok kezdő magánfelhasználónak is csalódást okozott a gép. Programozási nyelve – a BASIC 2.0 – meglehetősen struktúrálatlan. Grafikai és zenei lehetőségei – melyek pedig nagyban hozzájárultak elterjedéséhez – az alap BASIC-ből csak igen körülményesen érhetők el. A programnyelv e hiányosságainak, rossz tulajdonságainak ellensúlyozására készülnek újabb és újabb bővítések a C 64 BASIC-jére.

A legfontosabb ezek közül – kis túlzással „A BŐVÍTÉS"-nek nevezhető – a SIMON's BASIC. Általános célú; fő szempontja, hogy a gépet emberközelibbé tegye. Részterületek programozásához (pl. csak grafika) más bővítések inkább ajánlhatóak, de valamennyi közül a SIMON's a legbarátságosabb. Segítségével jól szervezett, áttekinthető programok írhatók, számos utasítás segíti a kényelmes programozást, és egyszerű parancsokkal vezérelhető a hang és a grafika is.

A DATA BECKER-sorozat legfrissebb, magyarul megjelent kötete e programnyelvet mutatja be. Kiemelkedő erénye a rendszerezettség, pontosság – sokkal inkább ajánlható a felhasználóknak, mint az eredeti SIMON's BASIC kézikönyv. (Hasznára vált a kötetnek, hogy a szerzők a kézikönyv figyelembe vétele nélkül írták meg, mivel az hemzseg a bizonytalanságoktól és ellentmondásoktól.) Felhívja a figyelmet a SIMON's BASIC fogyatékosságaira is – bíztatva ezáltal a felkészült programozókat azok kiküszöbölésére. E kötetből elsajátíthatóak a SIMON's nyelvű programírás jellegzetességei, technikája, a tankönyv-jelleget erősítik az egyes fejezetek végén olvasható összefoglaló feladatok is. A fordítás (az APEX GMK munkája) pontos, korrekt – csak ne szerepelne e kötetben is az elterjedt „string" fogalom helyett az élő beszédben soha nem használt „füzér". De egy kifejezés erőltetett magyarítása legyen a legnagyobb kifogás minden számítástechnikai témájú könyvvel szemben.

Tallér József

Gépnyerő

A SZUPER GÉPNYERŐ MEGOLDÁSA

Először is elnézést kell kérnünk, mert magyarázó példánkból félremagyarázó példa lett, ugyanis a listába 0,5 kg helyett 5 kg „íródott", s ráadásul az alatt lévő ábra nem követte a lista által meghatározott sorrendet. Reméljük olvasóink azért így is megértették a feladatot.

1. Tegyük fel, hogy a bőröndök ott fekszenek szépen sorban, s Benedek, illetve Rebeka már bepakolták a holmikat. Állítás: akárhogy is választunk ki egymás után 2 bőröndöt, a bennük található holmik összsúlya többₑ mint 6 kg. Ugyanis tekintsük azt a holmit, amit Benedek, ill. Rebeka először tett bele a 2 bőrönd közül a későbbibe. Ez a holmi a korábbi bőröndbe nem fért bele (sőt, egyik korábbiba se, hisz akkor oda került volna, az első olyanba, amelybe belefér), így tehát ennek a holminak a súlya + a korábbi bőröndben már ekkor megtalálható holmik összsúlya több, mint 6 kg, s később ez az összsúly csak növekedhet.

Ha a mama az L listát M(L) bőröndbe rakja, ez azt is jelenti, hogy a holmik összsúlya nem lehet több, mint M(L).6 kg. Ha Benedek vagy Rebeka ezeket legalább 2.M(L) bőröndbe rakná, az előző megállapításunk szerint (kettesével párosítva az első 2.M(L) db bőröndöt) azt jelentené, hogy a holmik összesúlya nagyobb M(L).6 kg-nál, ami nem lehet, tehát tényleg kevesebb, mint 2.M(L) bőröndbe pakolnak mind a ketten.

2. L1=(3,5 kg; 3,5 kg; 3 kg; 2 kg; 1,5 kg; 1,5 kg; 1 kg; 1 kg; 1 kg)

M(L1)=3, mert:

1. bőrönd: 3,5 kg; 1,5 kg; 1 kg
2. bőrönd: 3,5 kg; 1,5 kg; 1 kg
3. bőrönd: 3 kg; 2 kg; 1 kg

Mivel a súlyok sorban vannak, Benedek és Rebeka ugyanazt csinálják.

B(L1)=R(L1)=4, mert

1. bőrönd: 3,5 kg; 2 kg;
2. bőrönd: 3,5 kg; 1,5 kg; 1 kg
3. bőrönd: 3 kg; 1,5 kg; 1 kg
4. bőrönd: 1 kg

3. (Ezentúl a „kg"-okat elhagyjuk.)
L2=(3,5; 1,5; 1; 2; 2; 2)

M(L2)=B(L2)=2, mert

1. bőrönd: 3,5; 1,5; 1
2. bőrönd: 2; 2; 2

R(L2)=3, mert

1. bőrönd: 3,5; 2
2. bőrönd: 2; 2; 1,5
3. bőrönd: 1

4. L3=(6 db 0,8 kg-os; 6 db 2,1 kg-os; 6 db 3,1 kg-os)

M(L3)=R(L3)=6, nyilván mind a 6 bőröndben lesz 1–1 minden súlyból.

B(13)=10, mert először a 6 db 0,8kg-os holmit berakja az első bőröndbe, mivel itt csak 1,2 kg-nyi hely marad, ezt később nem használhatja. Ezután a 2,1 kg-os súlyokat pakolja be kettesével, ezzel újabb 3 bőröndöt „telít" meg. Végül a 6 db 3,1 kg-os súlyt kénytelen újabb 6 bőröndbe berakni.

Megjegyzések:

1. Természetesen ezek nem kizárólagos megoldások, bármilyen más jó megoldást is elfogadunk.
2. Szeretnénk néhány szót szólni arról, hogy mi értelme van ennek a feladatnak. Képzeljünk a bőröndök helyébe embereket, vagy mikroprocesszorokat, a holmik helyébe végrehajtandó feladatokat, a súlyok helyébe végrehajtási időket. Nyilván nem mindegy, hogy ugyanazt a feladathalmazt 6 vagy 10 ember végzi el egy nap alatt. Mama, Benedek és Rebeka helyébe pedig képzeljünk olyan programokat, amelynek meg kell határoznia, hogy ki (ill. melyik processzor) melyik feladatot kapja. Nyilván a mama programja lenne a legjobb, de egyelőre úgy néz nki, hogy senki se tudja (és valószínűleg nem is fogja tudni) a mama programját úgy megcsinálni, hogy az elég gyors legyen.

Hogy ez pontosan mit jelent, azt most nem részletezzük, egyszer majd egy hosszabb cikk keretében megmagyarázzuk. Most csak annyit, hogy a feladatban szereplő időarányok kevés elosztásra váró feladat esetén reálisak, ha azonban sok (pl. 1000) feladatot kell szétosztani, akkor a mama esetleg billiószor, vagy még többször annyi ideig dolgozik, mint pl. Rebeka. Ezért a gyakorlatban igen fontosak a Benedekéhez vagy Rebekáéhoz hasonló ún. szuboptimális algoritmusok, melyek nem a legjobb megoldást adják, de elég gyorsak. S ha már ilyeneket használnak, sokszor fontos az is, hogy meg tudjuk becsülni, ezek hányszor rosszabbak az optimális algoritmusnál. Jelen esetben láttuk, hogy mindkét gyerek mindig képes az optimum kétszeresénél kevesebbe pakolni, s ez már magában is nagy eredmény. Azonban azt is láttuk, hogy van olyan eset, hogy Rebeka az optimum másfélszeresébe.Benedek pedig az 5/3-szorosába pakol, tehát azért nem igazán jók ezek az algoritmusok. Az igazság a következő: minden lehetséges L listára B(L)<1,7.M(L)+1, azonban létezik végtelen sok L lista, hogy B(L)>1,7.M(L)−8. Másrészt minden L listára R(L)≤11/9.M(L)+4, de létezik végtelen sok lista, melyre R(L)≥11/9.M(L)−9. Látszik, hogy a csak kicsit lassúbb Rebeka mennyivel megbízhatóbb!

Irodalom: Lovács–Gács: Algoritmusok
Varga L.: Rendszerprogramok elmélete és gyakorlata

A Szuper BIT-LET gépnyerő kiértékelése még tart. Eredményeket jövő hónapban közlünk.

Bármely program bonyolultsága addig fokozódik, amíg túl nem nő programozója képességein!
(Murphy törvénykönyve)

harmadgép nyerő



Jó hírünk van. Az ÁPISZ-szal kötött együttműködési megállapodásnak köszönhetjük, hogy most egy Commodore +4-es gépért játszhatnak olvasóink. Harmadgépnyerőről van szó, mert három fordulóban lehet megnyerni a gépet. Pontversenyünkben a végső pontoknak megfelelően a legjobb 25 versenyzőnek van nyerési esélye. A sorsolásban az első tíz helyezett neve 3x, a második tízé 2x, az utolsó öt versenyzőé pedig 1x–1x kerül majd a sorsoió programba. Az első díjon kívül tíz további kisebb díjat is kisorsolunk. (Vásárlási utalványokat, ajándék csomagokat.)

Az első fordulóban 15 pontot lehet szerezni.

Mielőtt az új feladatokat olvasnák, íme néhány információ előző pályázatainkkal kapcsolatban.

1. A szuper BIT-LET gépnyerő megoldásainak értékelése még tart.

2. A C 16 nyerőre beküldött programokat még nem sikerült leküzdenünk. Türelmet kérünk. (A beérkezett programok egy része egészen profi!)

3. A beküldött kazettákat és disckeket természetesen visszaküldjük majd. (Most adtunk postára egy nagy kupac régebbi pályázat óta nálunk levő kazettát!)

Íme a Harmadgépnyerő első feladata:

Az előző oldalon közölt Szuper BIT-LET-feladat megoldásában láthattuk, hogy milyen sokat segíthet az, ha egy algoritmus végrehajtása előtt a feldolgozandó adatokat valahogy rendezzük. Ez sok más algoritmusnál is így megvan, sok más esetben pedig a rész- vagy végeredményt kell valamiért rendezni, így a rendezések az algoritmusok között bizonyos központi szerephez jutnak. A gyakorlat számára igen jelentős, hogy a rendezéseket gyorsan tudjuk elvégezni. Ezért van az, hogy a rendezésekkel kapcsolatban számos elméleti eredmény alakult ki (főleg az utóbbi néhány évtizedben). Most a rendezésnek egy igen különös, mégis a gyakorlatban is jelentős „fajtájával" ismerkedünk meg, melyet networknek hívnak. (Sajnos magyar neve még nem igazán alakult ki, akit zavar az angol szó, hívhatja hálózatnak, de úgy érezzük, ez a lefordítás nem elég szerencsés, így mi az angol szónál maradunk.) A networköt úgy kell elképzelni, hogy van valahány párhuzamosan futó drótunk, melyeken 1–1 szám „halad". Néhol összehasonlító kapuk vannak két drót között. Pl. legyen az első kapu a 2. és az 5. drót között. Ha a 2. dróton érkező szám nem nagyobb az 5. dróton érkezőnél, akkor a kapunál semmi sem történik, mindkét szám az eredeti helyén folytatja útját. Ha azonban a 2. dróton érkező szám a nagyobb, akkor helyet cserélnek, ez az 5. dróton, az ott érkező pedig a 2. dróton megy tovább. (1a–1b ábra) Egy network ilyen párhuzamos drótokból és köztük levő kapukból áll (2. ábra). Egy kapu után mindig a felső dróton megy tovább a kisebb szám (a 2 beérkező közül), az alsó dróton pedig a nagyobb (kivéve persze, ha egyformák). Világos, hogy egy ilyen networkkel lehet rendezni. Azt mondjuk, hogy egy network rendez, ha akárhogyan engedünk rá számokat a drótokra, azok a másik végén nagyság szerinti növekvő sorrendben érkeznek meg. Pl. a 2. ábrán látható network 4 számot képes rendezni. Hogyan dönthető el egy networkről, hogy rendező-e? Könnyű belátni, hogy ehhez elég, hogyha az n drótból álló networkre ráengedjük az 1, 2..., n számokat az összes lehetséges sorrendben, akkor mindig rendezve jönnek ki belőle. Ezt pedig ki lehet próbálni, igaz, hogy n! bemenetet kell ellenőrizni, ami nagy n-nél sajnos igen lassú. Ezen egy kicsit segít a következő

állítás: egy n huzalos network akkor és csakis akkor rendező, ha bármilyen n hosszúságú 0–1 sorozatot ráengedve, azt rendezi.

1. Feladat: Bizonyítsuk be ezt az állítást!

Ezzel elértük, hogy csak 2 n bemenetet kell ellenőrizni, sajnos ez majdnem ugyanolyan sok, már egy 10 drótos networknál is 1024 bemenet útját kell végigkövetnünk. Ezért a nagyobb networkök rendező voltát nem így állapítják meg, hanem eleve úgy készítik őket, hogy bizonyítható legyen: ők rendezők. Ezt általában úgy lehet elérni, hogy bizonyos funkciókat ellátó részeket raknak egymás után.

2. Feladat: Bizonyítsuk be, hogy a 3. ábrán látható network rendező.

Talán ennyiből is látható, hogy networkkel nehezebb rendezni, mint a szokásos módon, hiszen a valamely pontig végzett összehasonlítások során nyerhető információ nagy része elvész, ugyanis nem tudjuk, hogy régebben melyik számot melyikkel hasonlítottuk össze. Erre jó példa az, hogy ha pl. tudjuk valamelyik számról, hogy ő a középső, akkor a szokásos módszerekkel n lépésben kiválaszthatjuk a tőle kisebb számokat, networknél azonban ez az információ nem használható fel, hiszen néhány összehasonlítás után már nem tudjuk, hol van a mi középső számunk.

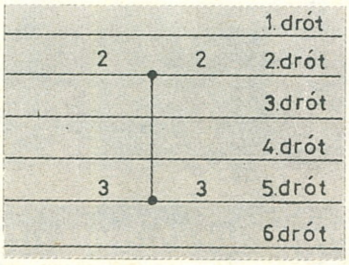
Networkök vizsgálatánál jól használható a következő egyszerű állítás: lehetetlen, hogy egy network végén egy kisebb sorszámú dróton mindig (értsd: az 1, 2..., n számok bármilyen permutációjával indítva) nagyobb érték érkezzen, mint egy nagyobb sorszámúra.

3. Feladat: Lássuk ezt be!

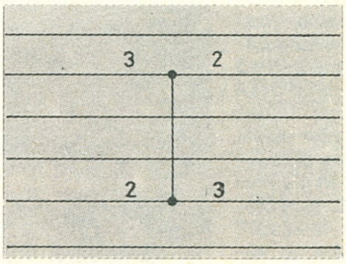
A networkök gyakorlati alkalmazásainál azonban legfontosabb az ún. párhuzamosíthatóságuk. Ez a fogalom azt takarja, hogy ha két kapu 4 különböző drótra illeszkedik, akkor egyszerre is elvégezhetik feladatukat. Nevezzük futamnak a kapuk olyan összességét, melyek mind különböző drótra támaszkodnak, tehát nincs olyan drót, amelyre egynél több kapu illeszkednék. Úgy tekintjük a dolgot, hogy az egy futamban lévő kapuk egyszerre dolgoznak. Így egy rendező network végrehajtási ideje nem a kapuk, hanem a futamok számától függ, ami általában sokkal kisebb. Pl. a 2. ábrán látható network 3 futamból áll, az első az első 2 kapuból, a 2. a második két kapuból, végül a 3. az utolsó kapuból áll.

4. Feladat: Készítsünk 5 futamból álló, 6 számot rendező networköt!

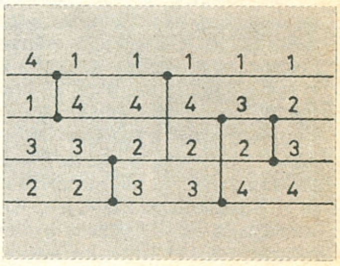
1/a. ábra



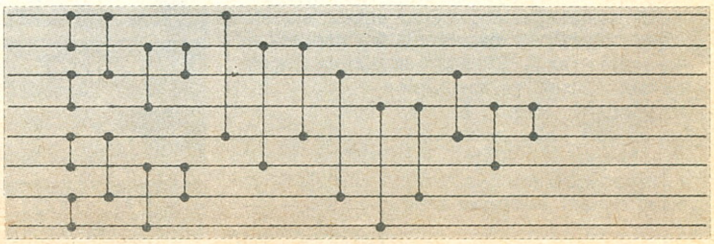
1/b. ábra



2. ábra



3. ábra



BIT-LET

Szerkesztőségünk címe: ÖTLET Budapest 198. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál

Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Gépnyerő: Király Zoltán. Posta: Halász Péter